

TUGAS AKHIR - KI141502

PENENTUAN HARGA DENGAN MENGGUNAKAN SISTEM INFERENSI FUZZY TSUKAMOTO PADA RANCANG BANGUN APLIKASI “FINDING-TUTOR”

SYAH DIA PUTRI MUSTIKA SARI
NRP 5113100191

Dosen Pembimbing I
Dr.tech. Ir. R.V.Hari Ginardi, M.Sc

Dosen Pembimbing II
Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



TUGAS AKHIR - KI141502

PENENTUAN HARGA DENGAN MENGGUNAKAN SISTEM INFERENSI FUZZY TSUKAMOTO PADA RANCANG BANGUN APLIKASI “FINDING-TUTOR”

**SYAH DIA PUTRI MUSTIKA SARI
NRP 5113100191**

**Dosen Pembimbing I
Dr.tech. Ir. R.V.Hari Ginardi, M.Sc**

**Dosen Pembimbing II
Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.**

**DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya, 2017**

[Halaman ini sengaja dikosongkan]



UNDERGRADUATE THESIS - KI141502

PRICING IN FINDING TUTOR APPLICATION USING TSUKAMOTO FUZZY INFERENCE SYSTEM

**SYAH DIA PUTRI MUSTIKA SARI
NRP 5113100191**

**Supervisor I
Dr.tech. Ir. R.V.Hari Ginardi, M.Sc**

**Supervisor II
Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.**

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
Sepuluh Nopember Institute of Technology
Surabaya, 2017**

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

PENENTUAN HARGA DENGAN MENGGUNAKAN SISTEM INFERENSI FUZZY TSUKAMOTO PADA RANCANG BANGUN APLIKASI “FINDING-TUTOR”

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Rumpun Mata Kuliah Manajemen Informasi
Program Studi S-1 Departemen Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

SYAH DIA PUTRI MUSTIKA SARI
NRP: 5113 100 191

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Dr.tech. Ir. R.V. Hari Ginardi, M.Sc.
NIP. 196505181992031003



(Pembimbing 1)

Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.
NIP. 197512202001122002

(Pembimbing 2)

SURABAYA
MEL, 2017

[Halaman ini sengaja dikosongkan]

PENENTUAN HARGA DENGAN MENGGUNAKAN SISTEM INFERENSI FUZZY TSUKAMOTO PADA RANCANG BANGUN APLIKASI FINDING TUTOR

Nama Mahasiswa : SYAH DIA PUTRI MUSTIKA SARI
NRP : 5113 100 191
Jurusan : Teknik Informatika, FTIf-ITS
Dosen Pembimbing 1: Dr. Tech. Ir. R.V.Hari Ginardi, M.Sc
**Dosen Pembimbing 2: Dr.Eng. Chastine Fatichah, S.Kom.,
M.Kom.**

Abstrak

Keterbatasan waktu yang dimiliki di sekolah membuat kegiatan belajar tidak hanya berlangsung di dalam ruang kelas saja. Orang tua juga tidak dapat kita libatkan secara maksimal dalam kegiatan belajar karena keterbatasan waktu dan kemampuan materi yang dikuasai. Hal ini menjadikan tutor sebagai salah satu profesi yang cukup dibutuhkan dan menjanjikan. Namun, keterbatasan waktu dan lokasi membuat tutor kesulitan menemukan konsumen yang sesuai. Selain itu, kendala utama yang dihadapi konsumen adalah ketetapan harga tutor yang cukup mahal.

Tugas akhir ini ditujukan untuk mengatasi masalah tersebut dengan sebuah aplikasi berbasis Android yang dapat membantu konsumen mencari tutor. Aplikasi tersebut dilengkapi dengan sebuah sistem pendukung keputusan yang dapat membantu menghitung harga layanan dengan Sistem Inferensi Fuzzy Tsukamoto. Hal ini memungkinkan system menghitung harga layanan dengan tepat berdasarkan dengan kriteria tertentu.

Hasil dari penelitian ini adalah berupa aplikasi pencarian tutor yang dilengkapi dengan Sistem Inferensi Fuzzy Tsukamoto yang menghitung harga layanan berdasarkan tingkat kesulitan, waktu dan jarak.

Kata kunci: Sistem Inferensi Fuzzy, Fuzzy Tsukamoto, Android.

[Halaman ini sengaja dikosongkan]

PRICING IN FINDING TUTOR APPLICATION USING TSUKAMOTO FUZZY INFERENCE SYSTEM

Student's Name : SYAH DIA PUTRI MUSTIKA SARI
Student's ID : 5113 100 191
Departement : Teknik Informatika, FTIf-ITS
First Advisor : Dr. Tech. Ir. R.V.Hari Ginardi, M.Sc
**SecondAdvisor : Dr.Eng. Chastine Fatichah, S.Kom.,
M.Kom.**

Abstract

Limitation of time at school make learning activities not only take place in the classroom. Parents also can't optimally involve in learning activities due their time limitation and ability to solve the problems. Therefore "tutor" (non-formal teacher) as professions that is quite needed and interesting for many peoples. But, the availability of time and the location between them is the main problem for tutor to find the right consumer. In addition, the other problem faced by consumers is the price of tutors that quite expensive.

This undergraduate thesis intend to solve the problems with an Android Mobile Application that can help consumers find a tutor or tutor to find their consumer. This application comes with a decision support system that can help to calculate the price of tutor with Tsukamoto Fuzzy Inference System. The support system allow the system to calculate the price based on certain criteria.

The results of this research is Android Mobile Application with a Tsukamoto Fuzzy Inference System that calculate tutor's price based on difficulty level, time and distance.

Keywords: Fuzzy Inference System, Fuzzy Tsukamoto, Android.

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji bagi Allah SWT yang telah melimpahkan rahmat dan anugerah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul **“Penentuan Harga Dengan Menggunakan Metode Fuzzy Tsukamoto Pada Rancang Bangun Aplikasi Finding Tutor”**.

Buku tugas akhir ini disusun dengan harapan dapat memberikan manfaat dalam penelitian *Fuzzy Inference System* lebih lanjut. Selain itu, penulis berharap dapat memberikan kontribusi positif bagi kampus Teknik Informatika ITS.

Dalam perancangan, pengerjaan, dan penyusunan tugas akhir ini, penulis banyak mendapatkan bantuan dari berbagai pihak. Penulis ingin mengucapkan terima kasih kepada:

1. Bapak Dr.tech. Ir. R.V. Hari Ginardi, M.Sc dan Ibu Dr.Eng. Chastine Fatichah, S.Kom., M.Kom. selaku dosen pembimbing penulis yang telah memberi ilmu, nasihat dan arahan sehingga penulis dapat menyelesaikan tugas akhir dengan tepat waktu.
2. Orang tua penulis Alm. Bapak Mustari dan Ibu Ninik K. M yang telah memberikan dukungan moral, spiritual dan material serta senantiasa memberikan doa demi kelancaran dan kemudahan penulis dalam mengerjakan tugas akhir.
3. Saudara kandung: adik Ilham Putra Arifa, calon pendamping hidup: Hadiri Imam Mawardi, keluarga dekat: Budhe Endang, Ibu Indina Zulfa, dan Budhe Sas serta seluruh keluarga besar yang telah memberikan dukungan yang besar baik secara langsung maupun secara implisit.
4. Teman satu bimbingan Pak Hari (Astidhita, Lino, Rifqi, Nyoman, Riska) dan teman satu bimbingan Bu Chastine (Lusiana) yang membantu dalam pengerjaan, mengingatkan dan memberikan semangat untuk bimbingan bersama.
5. Teman-teman seperti Fany dan Luffy yang selalu baik membantu dalam kesulitan pengerjaan Tugas Akhir. Teman-

teman lainnya seperti Andriana, Icha, Zaza, Fira, Dina, Yohana dan mahasiswa angkatan 2013 lain yang selalu mengisi hari-hari penulis dengan kebahagiaan dan semangat untuk segera menyelesaikan studi.

6. Pihak-pihak lain yang tidak bisa penulis sebutkan satu-persatu.

Penulis menyadari masih ada kekurangan dalam penyusunan tugas akhir ini. Penulis mohon maaf atas kesalahan, kelalaian maupun kekurangan dalam penyusunan tugas akhir ini. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan ke depan.

Surabaya, Mei 2017

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	Error! Bookmark not defined.
Abstrak	vii
Abstract	ix
KATA PENGANTAR.....	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL	xix
DAFTAR KODE SUMBER.....	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Tugas Akhir	2
1.5 Manfaat Tugas Akhir	2
1.6 Metodologi	3
1.7 Sistematika Laporan.....	4
BAB II DASAR TEORI.....	7
2.1 Penelitian Terkait	7
2.2 E-Commerce	8
2.3 Artificial Intelligence	9
2.4 Sistem Inferensi Fuzzy	10
2.4.1 Variabel Fuzzy.....	11
2.4.2 Semesta Pembicaraan	11
2.4.3 Domain Himpunan Fuzzy	12
2.4.4 Fungsi Keanggotaan	12
2.4.4 Himpunan Fuzzy.....	16
2.4.5 Operasi Himpunan Fuzzy	17
2.4.6 Fuzzyfikasi.....	19
2.4.7 Sistem Inferensi	21
2.4.8 Defuzzyfikasi	23
2.4.9 Metode Tsukamoto	24
2.5 Java.....	27

2.6	NetBeans IDE.....	27
2.7	Android Studio.....	28
BAB III ANALISIS DAN PERANCANGAN SISTEM.....		29
3.1	Tahap Analisis.....	29
3.1.1	Deskripsi Umum Sistem.....	29
3.1.2	Data.....	30
3.1.3	Analisis Permasalahan.....	32
3.2	Tahap Perancangan Sistem.....	32
3.2.1	Desain Umum Sistem.....	33
3.2.2	Perancangan Data.....	36
3.2.3	Perancangan Proses.....	40
BAB IV IMPLEMENTASI.....		49
4.1	Lingkungan Implementasi.....	49
4.2	Implementasi Metode pada NetBeans IDE.....	49
4.2.1	Implementasi Fungsi Keanggotaan.....	50
4.2.2	Implementasi Fuzzyfication dan Inference System.....	57
4.2.3	Implementasi Defuzzyfication.....	60
4.3	Implementasi Metode pada Android Studio.....	61
4.3.1	Implementasi Persiapan Masukan Sistem Fuzzy.....	61
4.3.2	Implementasi Fungsi Keanggotaan.....	65
4.3.3	Implementasi Fuzzyfication dan Inference System.....	72
4.3.4	Implementasi Deffuzzyfication.....	77
4.3.5	Implementasi Perhitungan Harga Akhir.....	77
4.4	Implementasi Website Statistik Proses Bisnis.....	78
4.4.1	Implementasi Halaman Utama (Dashboard).....	78
4.4.2	Implementasi Halaman Data Pendaftar Tutor.....	82
4.4.3	Implementasi Halaman Data Tutor Terbaik.....	83
4.4.4	Implementasi Halaman Data Pendaftar Murid.....	84
4.4.5	Implementasi Halaman Data Murid Terbaik.....	85
4.4.6	Implementasi Data Transaksi.....	87
BAB V UJI COBA DAN EVALUASI.....		89
5.1	Lingkungan Uji Coba.....	89
5.2	Data Uji Coba.....	89

5.3	Uji Coba	90
5.4	Evaluasi	109
BAB VI KESIMPULAN DAN SARAN		111
5.5	Kesimpulan	111
5.6	Saran.....	111
LAMPIRAN		113
DAFTAR PUSTAKA.....		115
BIODATA PENULIS.....		117

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 Ilustrasi Sistem Inferensi Fuzzy.....	11
Gambar 2.2 Representasi Fungsi Keanggotaan Linear Naik	13
Gambar 2.3 Representasi Fungsi Keanggotaan Linear Turun	13
Gambar 2.4 Representasi Fungsi Keanggotaan Segitiga	14
Gambar 2.5 Representasi Fungsi Keanggotaan Trapesium	15
Gambar 2.6 Representasi Fungsi Keanggotaan Gaussian.....	16
Gambar 2.7 Himpunan Fuzzy Berat Badan.....	16
Gambar 2.8 Irisan Fuzzy Set A dan B[3]	17
Gambar 2.9 Gabungan Fuzzy Set A dan B[3]	18
Gambar 2.10 Komplemen Fuzzy Set A[3]	19
Gambar 2.11 Fungsi Keanggotaan (Contoh Kasus)	20
Gambar 2.12 Fungsi Keanggotaan 2 (Contoh Kasus)	21
Gambar 2.13 Ilustrasi Metode Mamdani	24
Gambar 2.14 Ilustrasi Metode Sugeno	25
Gambar 2.15 Ilustrasi Metode Tsukamoto	26
Gambar 3.1 Diagram Alir Sistem.....	33
Gambar 3.2 Use Case Diagram Aplikasi Finding Tutor.....	34
Gambar 3.3 Use Case Diagram Website	34
Gambar 3.4 Activity Diagram Tutor	35
Gambar 3.5 Activity Diagram Murid	35
Gambar 3.6 Fungsi Keanggotaan Tingkat Kesulitan.....	36
Gambar 3.7 Fungsi Keanggotaan Waktu.....	37
Gambar 3.8 Fungsi Keanggotaan Jarak	38
Gambar 3.9 Fungsi Keanggotaan Harga.....	39
Gambar 3.10 Diagram Alir Admin.....	41
Gambar 3.11 Diagram Alir Tutor	43
Gambar 3.12 Diagram Alir Murid	45
Gambar 5.1 Tampilan (a) Estimasi Harga (b) Harga Akhir	90
Gambar 5.2 Tampilan Halaman Utama (Dashboard)	91
Gambar 5.3 Tampilan Representasi Grafik	92
Gambar 5.4 Tampilan Halaman Data Pendaftar Tutor	93
Gambar 5.5 Tampilan Halaman Data Tutor Terbaik.....	94
Gambar 5.6 Tampilan Halaman Data Pendaftar Murid	95
Gambar 5.7 Tampilan Halaman Data Murid Terbaik	96
Gambar 5.8 Tampilan Halaman Data Transaksi.....	97

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 3.1 Masukan Tingkat Kesulitan	31
Tabel 3.2 Kategori Harga Durasi 90 Menit.....	31
Tabel 3.3 Daftar Zona Daerah.....	32
Tabel 3.4 Aturan (Rule) Perhitungan Estimasi Harga.....	47
Tabel 3.5 Aturan (Rule) Perhitungan Harga	48
Tabel 4.1 Lingkungan Perancangan Perangkat Lunak.....	49
Tabel 5.1 Tabel Uji Coba UC.001.....	98
Tabel 5.2 Tabel Uji Coba UC. 002.....	99
Tabel 5.3 Kategori Harga Durasi 90 Menit.....	99
Tabel 5.4 Hasil Uji Coba Perhitungan Estimasi Harga	100
Tabel 5.5 Hasil Uji Coba Perhitungan Harga Akhir	102
Tabel 5.6 Hasil Uji Coba Keseluruhan.....	110
Tabel 5.7 Hasil Uji Coba Keseluruhan.....	110

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 4.1 Implementasi Fungsi Keanggotaan (1.1).....	50
Kode Sumber 4.2 Implementasi Fungsi Keanggotaan (1.2).....	51
Kode Sumber 4.3 Implementasi Fungsi Keanggotaan (1.3).....	51
Kode Sumber 4.4 Implementasi Fungsi Keanggotaan (2.1).....	52
Kode Sumber 4.5 Implementasi Fungsi Keanggotaan (2.2).....	52
Kode Sumber 4.6 Implementasi Fungsi Keanggotaan (2.3).....	53
Kode Sumber 4.7 Implementasi Fungsi Keanggotaan (3.1).....	54
Kode Sumber 4.8 Implementasi Fungsi Keanggotaan (3.2).....	54
Kode Sumber 4.9 Implementasi Fungsi Keanggotaan (3.3).....	55
Kode Sumber 4.10 Implementasi Fungsi Keanggotaan (4.1).....	55
Kode Sumber 4.11 Implementasi Fungsi Keanggotaan (4.1.1)....	56
Kode Sumber 4.12 Implementasi Fungsi Keanggotaan (4.2).....	56
Kode Sumber 4.13 Implementasi Fungsi Keanggotaan (4.2.1)....	56
Kode Sumber 4.14 Implementasi Fungsi Keanggotaan (4.3).....	57
Kode Sumber 4.15 Implementasi Fungsi Keanggotaan (4.3.1)....	57
Kode Sumber 4.16 Implementasi Fungsi <code>hitungX()</code> (1)	58
Kode Sumber 4.17 Implementasi Fungsi <code>hitungX()</code> (2)	59
Kode Sumber 4.18 Implementasi Fungsi <code>hitungY()</code> (1)	59
Kode Sumber 4.19 Implementasi Fungsi <code>hitungY()</code> (2)	60
Kode Sumber 4.20 Implementasi Fungsi <code>defuzzyfikasi()</code>	60
Kode Sumber 4.21 Implementasi Masukan 1 (1)	61
Kode Sumber 4.22 Implementasi Masukan 1 (2)	62
Kode Sumber 4.23 Implementasi Masukan Waktu	63
Kode Sumber 4.24 Implementasi Masukan Jarak	64
Kode Sumber 4.25 Implementasi Fungsi Keanggotaan (1.1).....	65
Kode Sumber 4.26 Implementasi Fungsi Keanggotaan (1.2).....	66
Kode Sumber 4.27 Implementasi Fungsi Keanggotaan (1.3).....	66
Kode Sumber 4.28 Implementasi Fungsi Keanggotaan (2.1).....	67
Kode Sumber 4.29 Implementasi Fungsi Keanggotaan (2.2).....	67
Kode Sumber 4.30 Implementasi Fungsi Keanggotaan (2.3).....	68
Kode Sumber 4.31 Implementasi Fungsi Keanggotaan (3.1).....	69
Kode Sumber 4.32 Implementasi Fungsi Keanggotaan (3.2).....	69
Kode Sumber 4.33 Implementasi Fungsi Keanggotaan (3.3).....	70

Kode Sumber 4.34 Implementasi Fungsi Keanggotaan (4.1).....	70
Kode Sumber 4.35 Implementasi Fungsi Keanggotaan (4.1.1)...	71
Kode Sumber 4.36 Implementasi Fungsi Keanggotaan (4.2).....	71
Kode Sumber 4.37 Implementasi Fungsi Keanggotaan (4.2.1)...	71
Kode Sumber 4.38 Implementasi Fungsi Keanggotaan (4.3).....	72
Kode Sumber 4.39 Implementasi Fungsi Keanggotaan (4.3.1)...	72
Kode Sumber 4.40 Implementasi Fungsi <code>hitungX()</code> (1)	73
Kode Sumber 4.41 Implementasi Fungsi <code>hitungX()</code> (2)	74
Kode Sumber 4.42 Implementasi Fungsi <code>hitungY()</code> (1)	74
Kode Sumber 4.43 Implementasi Fungsi <code>hitungY()</code> (2)	75
Kode Sumber 4.44 Implementasi <code>hitungX()</code> Estimasi Harga .	76
Kode Sumber 4.45 Implementasi <code>hitungY()</code> Estimasi Harga .	76
Kode Sumber 4.46 Implementasi Fungsi <code>defuzzyfikasi()</code> ..	77
Kode Sumber 4.47 Implementasi Perhitungan Harga Akhir	78
Kode Sumber 4.48 Implementasi Pendaftaran Tutor Hari Ini	79
Kode Sumber 4.49 Implementasi Pendaftaran Murid Hari Ini.....	79
Kode Sumber 4.50 Implementasi Transaksi Hari Ini	80
Kode Sumber 4.51 Implementasi Jumlah Pemasukan Hari Ini ...	80
Kode Sumber 4.52 Implementasi Grafik Transaksi Per-Bulan ...	81
Kode Sumber 4.53 Implementasi Grafik Transaksi Per-Tahun...	81
Kode Sumber 4.54 Implementasi Data Pendaftaran Tutor.....	82
Kode Sumber 4.55 Implementasi Data Tutor Terbaik.....	84
Kode Sumber 4.56 Implementasi Data Pendaftaran Murid.....	Error!
Bookmark not defined.	
Kode Sumber 4.57 Implementasi Data Murid Terbaik	86
Kode Sumber 4.58 Implementasi Data Transaksi (1).....	87
Kode Sumber 4.59 Implementasi Data Transaksi (2).....	88

BAB I

PENDAHULUAN

Pada bab ini dibahas hal-hal yang mendasari tugas akhir. Bahasan meliputi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika laporan tugas akhir. Diharapkan dari penjelasan dalam bab ini gambaran tugas akhir secara umum dapat dipahami.

1.1 Latar Belakang

Pentingnya kegiatan belajar tidak hanya berlangsung di dalam ruang kelas saja, tetapi juga di luar lingkungan sekolah. Tidak sedikit siswa yang paham dengan materi yang telah disampaikan di sekolah, namun kesulitan saat mengerjakan soal. Orang tua juga tidak dapat kita libatkan secara maksimal dalam kegiatan belajar karna keterbatasan waktu dan kemampuan materi yang dikuasai. Selain itu, keinginan masyarakat untuk menambah kemampuan mereka dalam berbagai bidang juga sudah menjadi kebutuhan yang umum. Maka dari itu, salah satu solusi untuk menangani masalah seperti ini adalah dengan mendatangkan guru les privat (tutor) untuk membantu proses belajar mengenai berbagai materi di sekolah maupun kemampuan-kemampuan di bidang lain.

Pada saat ini permintaan guru les privat (tutor) sudah semakin besar dan akan terus bertambah. Namun, masyarakat harus tetap jeli dalam memilih tutor yang aman dengan harga yang bersaing. Banyaknya kriteria yang menentukan kualitas keamanan tutor membuat harga yang ditawarkan sangat bervariasi. Maka dari itu, masyarakat membutuhkan sebuah wadah yang dapat memudahkan dalam mendapatkan guru tutor sesuai dengan keahlian yang dibutuhkan serta keamanan dan harga yang bersaing.

Seiring dengan perkembangan teknologi saat ini, adanya sebuah aplikasi yang merupakan teknologi yang sedang berkembang pesat pada saat ini dapat menjadi solusi dari permasalahan tersebut. Maka dari itu aplikasi *E-Commerce* berbasis perangkat bergerak (*mobile*) yang diberi nama “Finding-Tutor” akan dibuat.

Aplikasi ini dapat memudahkan masyarakat untuk mendapatkan tutor yang sesuai kebutuhan dan waktu dengan harga yang bersaing. Harga yang bersaing dapat digunakan *developer* untuk menarik masyarakat sebagai konsumen untuk menggunakan aplikasi ini. Data keluaran dari metode yang digunakan berupa harga yang merepresentasikan hasil dari perhitungan sistem dengan metode *Fuzzy Tsukamoto*. Untuk mendapatkan harga final yang akan digunakan maka data tersebut akan diolah kembali dengan masukan konsumen berupa durasi dan ketentuan zona-zona daerah yang sudah ditetapkan sebelumnya untuk menjadi harga yang menarik dan tetap menguntungkan untuk kedua belah pihak.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini adalah bagaimana implementasi metode *Fuzzy Tsukamoto* dalam perhitungan harga tutor pada aplikasi *Finding Tutor*.

1.3 Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan antara lain:

1. Permasalahan yang akan dibahas adalah mengenai penggunaan metode *Fuzzy Tsukamoto* dalam perhitungan harga tutor.
2. Variabel yang digunakan adalah variabel tingkat kesulitan, waktu, jarak dan harga.
3. Jangkauan aplikasi hanya hanya pada area Surabaya.
4. Aplikasi yang dibuat adalah aplikasi berbasis perangkat bergerak (*mobile*) *Android* dengan bahasa pemrograman Java.

1.4 Tujuan Tugas Akhir

Tujuan dari pembuatan tugas akhir ini adalah mendapatkan harga yang dapat bersaing di masyarakat dari hasil perhitungan harga dengan metode *Fuzzy Tsukamoto* yang akan digunakan pada aplikasi *E-Commerce* bernama "*Finding-Tutor*".

1.5 Manfaat Tugas Akhir

Hasil dari tugas akhir ini diharapkan dapat mempermudah masyarakat mendapatkan guru les privat (tutor) yang sesuai dengan

kebutuhannya dan harga yang bersaing yang juga dapat dimanfaatkan sebagai daya tarik aplikasi “*Finding-Tutor*”.

1.6 Metodologi

Tahapan-tahapan yang dilakukan dalam pengerjaan tugas akhir ini adalah sebagai berikut:

1. Penyusunan Proposal Tugas Akhir

Tahap awal yang dilakukan dalam pengerjaan tugas akhir ini adalah penyusunan proposal tugas akhir. Di dalam proposal diajukan Metode *Fuzzy Tsukamoto* sebagai metode yang digunakan pada perhitungan harga tutor.

2. Studi Literatur

Pada tahap ini dilakukan pencarian, pengumpulan, pemahaman, dan pembelajaran literatur yang berhubungan dengan hal-hal yang mempengaruhi harga tutor pada umumnya, harga tutor pada pasaran, Metode *Fuzzy Tsukamoto* serta pemrograman *Java* pada *Android*. Studi literatur yang digunakan meliputi; hasil kuisioner dari penyedia jasa maupun pengguna jasa, buku referensi, jurnal, materi dalam suatu mata kuliah yang berhubungan dengan metode yang digunakan dan dokumen lainnya yang diperoleh dari internet.

3. Analisis dan Desain Perangkat Lunak

Pada tahap ini dilakukan analisa dari metode yang telah didapatkan dari studi literatur serta perancangan baru hasil dari analisa metode yang dikembangkan.

4. Implementasi Perangkat Lunak

Pada tahap ini dilakukan implementasi percobaan dalam program sederhana *Java* pada *NetBeans IDE* dengan menggunakan rancangan yang telah dibuat sebelumnya. Kemudian dilakukan implementasi hasil percobaan kedalam aplikasi “*Finding Tutor*” dengan menggunakan bahasa pemrograman *Java* pada *Android Studio*.

5. Uji Coba dan Evaluasi

Pada tahap ini dilakukan uji coba terhadap perangkat lunak yang telah dibuat untuk mengetahui kemampuan metode yang dipakai, mengamati kinerja sistem, serta mengidentifikasi kendala yang mungkin timbul. Parameter yang diuji cobakan adalah harga yang didapatkan dari perhitungan Metode *Fuzzy Tsukamoto* yang telah diolah kembali dengan ketentuan zona daerah yang telah ditetapkan .

6. Penyusunan Laporan Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan pengerjaan tugas akhir yang berisi dasar teori, dokumentasi dari metode yang dilakukan, dan hasil dari pengerjaan tugas akhir.

1.7 Sistematika Laporan

Buku tugas akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan tugas akhir ini. Selain itu, diharapkan dapat berguna bagi pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku tugas akhir terdiri atas beberapa bagian seperti berikut:

Bab I Pendahuluan

Bab yang berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan tugas akhir. Selain itu permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

Bab II Tinjauan Pustaka

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan tugas akhir ini.

Bab III Analisis dan Perancangan

Bab ini membahas mengenai perancangan perangkat lunak. Perancangan Alur Sistem, Perancangan perangkat lunak meliputi perancangan data, arsitektur, proses dan perancangan antarmuka pada sistem.

Bab IV Implementasi

Bab ini membahas mengenai implementasi dari rancangan yang telah dibuat pada bab sebelumnya. Penjelasan berupa *code* yang digunakan sebagai proses implementasi dan juga penunjang fitur dalam aplikasi.

Bab V Uji Coba dan Evaluasi

Bab ini menjelaskan mengenai kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari perangkat lunak yang telah dibuat sesuai dengan data yang diujikan.

Bab VI Kesimpulan dan Saran

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang telah dilakukan dan saran untuk pengembangan aplikasi ke depannya.

Daftar Pustaka

Merupakan daftar referensi yang digunakan dalam pembuatan tugas akhir.

Lampiran

Merupakan daftar referensi yang digunakan dalam pembuatan tugas akhir.

[Halaman ini sengaja dikosongkan]

BAB II

DASAR TEORI

Pada bab ini diuraikan mengenai dasar-dasar teori yang digunakan dalam pengerjaan tugas akhir dengan tujuan untuk memberikan gambaran secara umum terhadap penelitian yang dikerjakan. Bab ini berisi penjelasan mengenai *E-Commerce* dan *Artificial Intelligence* sebagai ruang lingkup dan studi kasus dari aplikasi yang akan dibangun, metode *Fuzzy Tsukamoto* sebagai metode yang akan digunakan pada proses perhitungan harga tutor, serta bahasa Java dan *Android Studio* sebagai bahasa pemrograman dan *tools* yang digunakan dalam pembuatan aplikasi.

2.1 Penelitian Terkait

Penelitian terkait yang telah dilakukan dengan metode *Fuzzy Tsukamoto* diantara lain yaitu “*Rainfall prediction in Tengger region Indonesia using Tsukamoto fuzzy inference system*”.[1] Pada penelitian tersebut metode *Fuzzy Tsukamoto* digunakan untuk membuat prediksi hujan pada daerah Tengger dengan menggunakan 4 kriteria berdasarkan dari sistem pakar. Selain itu juga terdapat penelitian dengan judul “Penerapan Logika *Fuzzy* dalam Menentukan Harga Jual Jamur Tiram Putih Berdasarkan Kualitas dan Permintaan Konsumen” oleh Kharisma Nara As’ad, Malikatul Munandiroh dan Marisa Fitri F dimana kriteria penentuan harga jamur berdasarkan dari tingkat kualitas jamur, harga jual jamur dan jumlah permintaan. Pada kedua penelitian tersebut metode *Fuzzy Tsukamoto* dapat diimplementasikan dengan baik sesuai dengan penelitian yang dilakukan. Metode ini dikenal sederhana karena hanya menggunakan 1 operasi himpunan dan proses defuzzyfikasi yang sederhana.

Pada penelitian ini, metode *Fuzzy Tsukamoto* diimplementasikan menggunakan 3 variabel karakteristik harga yaitu tingkat kesulitan, waktu dan jarak. Dengan menggunakan metode ini perhitungan harga tutor dapat dilakukan dengan berdasarkan persepsi dari masyarakat. Penggunaan metode ini juga membuat waktu komputasi lebih cepat karena perhitungannya dilakukan secara sederhana dan sesuai dengan karakteristik yang diinginkan.

2.2 *E-Commerce*

Bisnis yang berkembang pada saat ini lebih banyak memanfaatkan konsep *e-business*. *E-business* merupakan penggunaan kombinasi dari teknologi internet dan teknologi lain untuk bekerja dan mendukung proses bisnis yang ada pada suatu perusahaan.[2] Seiring dengan adanya perkembangan teknologi dan penggunaan konsep *e-business* pada saat ini, banyak perusahaan yang telah menerapkan konsep dari *e-commerce* dalam proses bisnis yang ada pada perusahaan tersebut.

E-commerce adalah suatu bentuk perubahan persaingan dalam seluruh proses interaksi bisnis seperti pemasaran, penjualan, pengiriman, pembayaran dan layanan pada transaksi global yang saling terkait.[1] *E-commerce* tidak hanya mencakup interaksi antara pelanggan untuk perusahaan dan perusahaan ke pemasok produk yang ditawarkan kepada pelanggan. Pada perusahaan yang telah melibatkan *e-commerce* akan memanfaatkan aplikasi layanan *E-Commerce* untuk menyelesaikan berbagai proses bisnis tersebut. Terdapat 4 kategori dasar pada aplikasi layanan *E-Commerce* yaitu *Business-to-Consumer* (B2C), *Business-to-Business* (B2B), *Consumer-to-Consumer* (C2C) dan *Business-to-Government* (B2G).[2]

Pada layanan *Business-to-Consumer* (B2C) bisnis akan dikembangkan dalam bentuk pasar online yang menarik untuk menjual produk dan melayani konsumen. Contoh layanan *Business-to-Consumer* (B2C) yang sering kali kita temui adalah *e-commerce website* maupun *mobile application* perusahaan yang menyediakan etalase secara virtual, katalog yang menarik, proses order yang interaktif, sistem pembayaran elektronik yang aman serta dukungan pelanggan secara *online*. Berbeda dengan layanan B2C yang menghubungkan proses bisnis perusahaan langsung kepada pelanggan, layanan pada *Business-to-Business* (B2B) merupakan interaksi antara dua perusahaan yang akan menghubungkan pasar *e-business* dan hubungan langsung yang ada pada proses bisnis tersebut. Contohnya saja pada perusahaan yang menawarkan *e-commerce website* yang aman untuk pelanggan bisnis maupun pemasok mereka. Selain kedua layanan yang telah dijelaskan diatas terdapat jenis

layanan lain yang meraih keberhasilan cukup besar yaitu layanan *Consumer-to-Consumer* (C2C). Pada layanan ini kedua aktor yang saling berinteraksi dapat berpartisipasi sebagai penjual maupun pembeli. Salah satu contoh dari layanan *Consumer-to-Consumer* (C2C) yang cukup dikenal oleh dunia yaitu *E-Bay*. [2]

Keuntungan yang didapatkan dari kesuksesan penerapan konsep *e-commerce* adalah memungkinkan berbagai proses bisnis yang ada pada perusahaan untuk dilakukan dimanapun, kapanpun dan dengan siapa saja secara lebih efisien. Faktor penting yang akan mempengaruhi kesuksesan pada penerapan *e-commerce* itu sendiri ada 6 yaitu pemilihan dan nilai dari produk, performa dan layanan pada proses pembelian serta pengiriman, tampilan dan kemudahan pada aplikasi *e-commerce*. [2]

2.3 Artificial Intelligence

Seiring banyaknya penerapan *e-commerce* pada perusahaan bisnis, teknologi *Artificial Intelligence* (AI) sering digunakan untuk mendukung maupun memperbaiki dukungan keputusan perusahaan. Teknologi AI merupakan sebuah bidang ilmu pengetahuan dan teknologi berdasarkan ilmu disiplin seperti ilmu komputer, biologi, psikologi, linguistik, matematika dan rekayasa. [2] Salah satu tujuan dari teknologi AI adalah untuk menduplikat kemampuan manusia untuk berfikir, melihat, mendengar, berjalan, berbicara dan merasakan pada sebuah komputer yang akan dikembangkan.

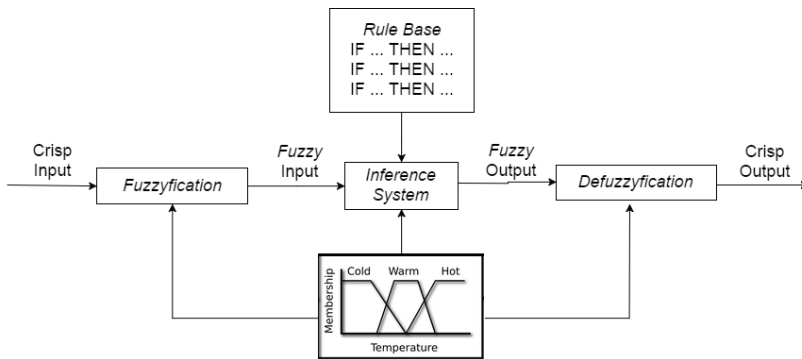
Aplikasi *Artificial Intelligence* (AI) memiliki 3 domain utama yaitu *Cognitive Science*, *Robotics*, dan *Natural Interfaces*. [2] Pada domain *Cognitive Science* berfokus pada penelitian bagaimana otak manusia dapat bekerja, berfikir dan belajar. Aplikasi pada domain ini meliputi pengembangan sistem pakar (*Expert System*), *adaptive learning systems*, *Fuzzy Logic*, *Genetic Algorithms*, *Neural Networks*, *Intelligent Agents* dan *knowledge-based system* lainnya. [2] Pada domain *Robotics*, AI berdiri sebagai ilmu disiplin dasar pada teknologi robot. Teknologi ini akan menghasilkan mesin robot dengan kecerdasan, sistem kontrol dan kemampuan fisik seperti manusia. Sebagai contohnya yaitu kemampuan robot untuk bergerak dan menemukan jalan menuju tujuan tertentu. Berbeda pada kedua domain

sebelumnya, pada domain *Natural Interfaces* hanya difokuskan pada pengembangan bahasa alami dan pengenalan suara. Contohnya saja seperti kemampuan komputer maupun robot untuk memahami apa yang kita katakan kepada mereka dengan *human natural languages*. [2]

Salah satu aplikasi AI yang diterapkan secara luas dalam bisnis adalah pengembangan sistem pakar (*Expert System*). Sistem pakar (*Expert System*) merupakan sistem informasi berbasis pengetahuan yang menggunakan pengetahuan tersebut pada aplikasi dan bertindak sebagai konsultan ahli bagi *end user*. [2] Sistem pakar dirancang untuk menyimpan maupun menyediakan jawaban untuk menjawab pertanyaan pada area masalah yang spesifik dengan membuat kesimpulan manusiawi tentang pengetahuan yang terdapat dalam basis pengetahuan khusus. Pengetahuan yang digunakan dalam sistem pakar merupakan aturan maupun informasi berbentuk premis dan kesimpulan seperti *IF* (kondisi), *THEN* (kesimpulan). [2] Terdapat 4 cara yang dapat digunakan untuk merepresentasikan pengetahuan ke dalam basis pengetahuan sistem pakar yaitu *Case-Based Reasoning* dimana basis pengetahuan berasal dari suatu kasus sebelumnya maupun pengalaman yang pernah ada, *Frame-Based Knowledge* dimana pengetahuan direpresentasikan dalam bentuk hirarki, *Object-Based Knowledge* dimana pengetahuan direpresentasikan sebagai sebuah jaringan objek yang mengandung data dan metode atau proses dari data tersebut dan *Rule-Based Knowledge* dimana pengetahuan direpresentasikan dalam bentuk aturan dan pernyataan fakta. [2]

2.4 Sistem Inferensi *Fuzzy*

Sistem Inferensi *Fuzzy* merupakan metode penalaran yang menyerupai penalaran manusia yang memungkinkan untuk memperkirakan nilai maupun kesimpulan dari data yang tidak lengkap atau data ambigu tanpa hanya mengandalkan crisp data, seperti yang terjadi pada data biner yang berisikan pilihan Iya/Tidak. [2] Dalam metode ini suatu preposisi dapat direpresentasikan dalam derajat kebenaran (truthfulness) atau kesalahan (falsehood) tertentu. Ilustrasi sistem inferensi *Fuzzy* dapat ditunjukkan pada **Gambar 2.1**.



Gambar 2.1 Ilustrasi Sistem Inferensi Fuzzy

Dapat dilihat pada ilustrasi sistem inferensi *fuzzy* diatas bahwa masukan berupa *crisp data* akan diolah menjadi data *fuzzy* sebagai masukan yang dibutuhkan sistem inferensi *fuzzy*. Data *fuzzy* tersebut kemudian akan diolah ke dalam sistem inferensi yang menyimpan aturan-aturan dalam bentuk premis yang akan digunakan dalam proses penarikan kesimpulan dari data masukan yang ada. Hasil dari proses penarikan kesimpulan tersebut berupa keluaran *fuzzy* yang kemudian akan diolah kembali menjadi *crisp data*. *Crisp data* tersebut merupakan keluaran akhir dari sistem inferensi *fuzzy* yang dapat dimanfaatkan dalam proses selanjutnya yang ada pada sistem.

2.4.1 Variabel Fuzzy

Variabel *Fuzzy* merupakan variabel yang akan dibahas dalam suatu sistem *Fuzzy*. Pada variabel yang digunakan pada sistem *Fuzzy* menggunakan konsep variabel linguistik.[3] Variabel linguistik merupakan variabel yang memiliki nilai berupa kata/kalimat. Contoh variabel *Fuzzy* yang sering digunakan yaitu umur, berat badan, tinggi badan dan lain sebagainya.

2.4.2 Semesta Pembicaraan

Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *Fuzzy*. Nilai semesta pembicaraan dapat berupa bilangan positif maupun negatif yang merupakan atribut numeris dari suatu variabel. Sebagai contoh

pada **Gambar 2.7** semesta pembicaraan atau nilai numeris untuk variabel berat badan adalah $[0,150]$ dimana variabel berat badan memiliki nilai 0 kg hingga 150 kg.

2.4.3 Domain Himpunan *Fuzzy*

Domain himpunan *Fuzzy* adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu himpunan *Fuzzy*. Nilai domain dapat berupa bilangan positif maupun negatif. Sebagai contoh dari domain himpunan *Fuzzy* berat badan pada **Gambar 2.7** dengan himpunan *Fuzzy* dan semesta pembicaraan yang telah dijelaskan seperti diatas yaitu RINGAN $[0,65]$ dimana berat badan diantara 0 hingga 65 kg dikatakan ringan, SEDANG $[50,80]$ dimana berat badan diantara 50 hingga 80 kg dikatakan sedang dan BERAT $[65,150]$ dimana berat badan diantara 65 hingga 150 kg dapat berat.

2.4.4 Fungsi Keanggotaan

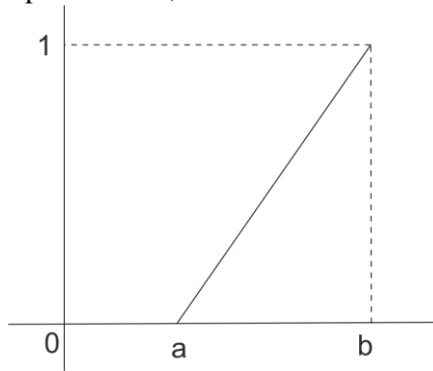
Fungsi keanggotaan adalah grafik yang mewakili besar dari derajat keanggotaan masing-masing variabel input yang berada dalam interval 0 dan 1. Derajat keanggotaan sebuah variabel x dilambangkan dengan simbol $\mu(x)$. Salah satu cara untuk mendapatkan nilai keanggotaan adalah dengan menggunakan pendekatan fungsi.[4] Beberapa fungsi keanggotaan yang dapat digunakan antara lain :

2.4.4.1 Fungsi Keanggotaan Linear Naik dan Turun

Fungsi keanggotaan linear naik maupun turun memiliki parameter a dan b . Untuk persamaan fungsi keanggotaan linear naik dapat ditunjukkan pada persamaan (2.1) seperti berikut:

$$\mu(x; a, b) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & x > b \end{cases} \quad (2.1)$$

Representasi grafik dari persamaan (2.1) dapat ditunjukkan pada **Gambar 2.2** seperti berikut,

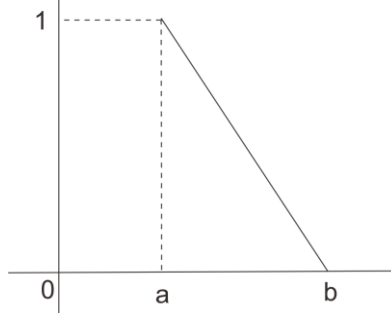


Gambar 2.2 Representasi Fungsi Keanggotaan Linear Naik

Untuk persamaan fungsi keanggotaan linear turun dapat ditunjukkan pada persamaan (2.2) seperti berikut:

$$\mu(x; a, b) = \begin{cases} 1, & x < a \\ \frac{b-x}{b-a}, & a \leq x \leq b \\ 0, & x > b \end{cases} \quad (2.2)$$

Representasi grafik dari persamaan (2.2) dapat ditunjukkan pada **Gambar 2.3** seperti berikut,



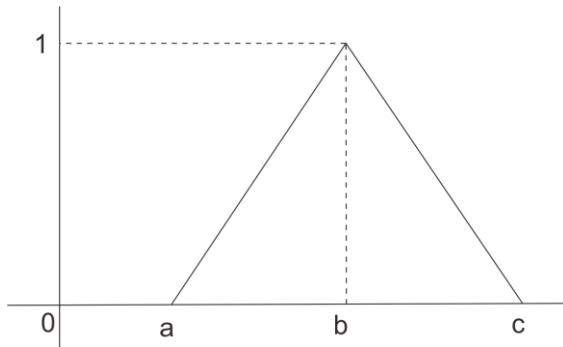
Gambar 2.3 Representasi Fungsi Keanggotaan Linear Turun

2.4.4.2 Fungsi Keanggotaan Segitiga

Fungsi keanggotaan segitiga memiliki parameter a, b, c dengan persamaan fungsi keanggotaan segitiga yang ditunjukkan pada persamaan (2.3) seperti berikut:[5]

$$\mu(x; a, b, c) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b < x \leq c \\ 1, & x > c \end{cases} \quad (2.3)$$

Representasi grafik dari persamaan (2.3) dapat ditunjukkan pada **Gambar 2.4** seperti berikut,



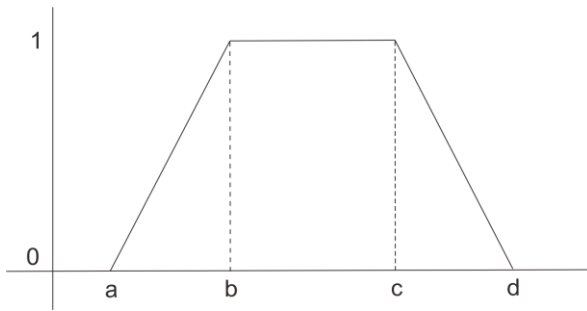
Gambar 2.4 Representasi Fungsi Keanggotaan Segitiga

2.4.4.3 Fungsi Keanggotaan Trapesium

Fungsi keanggotaan trapesium memiliki parameter a, b, c, d dengan persamaan fungsi keanggotaan trapesium yang ditunjukkan pada persamaan (2.4) seperti berikut:[5]

$$\mu(x; a, b, c, d) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b < x < c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & x > d \end{cases} \quad (2.4)$$

Representasi grafik dari persamaan (2.4) dapat ditunjukkan pada **Gambar 2.5** seperti berikut,



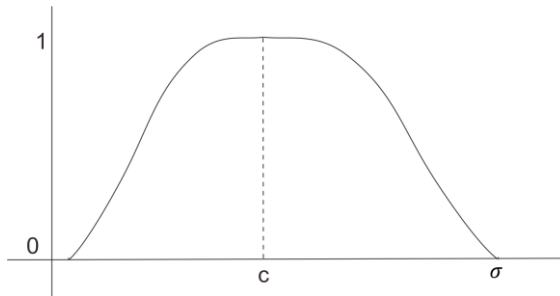
Gambar 2.5 Representasi Fungsi Keanggotaan Trapesium

2.4.4.4 Fungsi Keanggotaan Kurva Gaussian

Fungsi keanggotaan *Gaussian* memiliki parameter c dan σ dengan persamaan fungsi keanggotaan *Gaussian* yang ditunjukkan pada persamaan (2.5) seperti berikut:

$$\mu(x; c, \sigma) = \frac{1}{1 + \left(\frac{x-c}{\sigma}\right)^2} \quad (2.5)$$

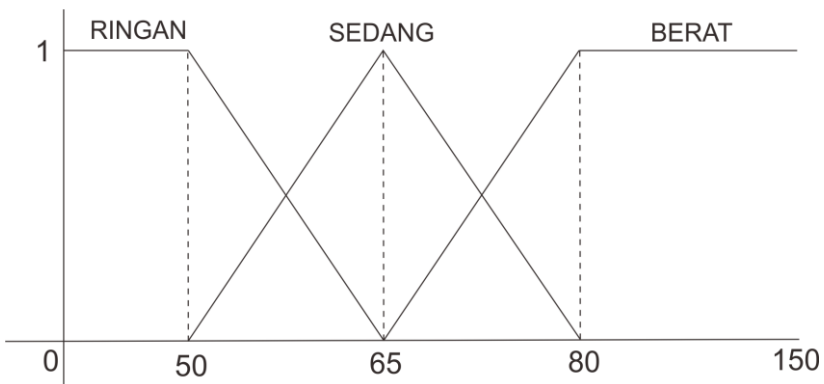
Representasi grafik dari persamaan (2.5) dapat ditunjukkan pada **Gambar 2.6** seperti berikut,



Gambar 2.6 Representasi Fungsi Keanggotaan *Gaussian*

2.4.4 Himpunan *Fuzzy*

Himpunan *Fuzzy* merupakan suatu kelompok yang mewakili suatu keadaan tertentu dalam suatu variabel *Fuzzy*. Representasi himpunan *Fuzzy* pada komputer berupa fungsi keanggotaan dengan nilai keanggotaan berupa interval yang telah ditetapkan sebelumnya. Dapat dilihat pada **Gambar 2.7** untuk contoh variabel *Fuzzy* yang digunakan yaitu berat badan maka himpunan *Fuzzy* atau nilai linguistik variabel tersebut adalah ringan, sedang, berat dengan interval nilai yang telah ditentukan sebelumnya.



Gambar 2.7 Himpunan *Fuzzy* Berat Badan

Setelah membentuk fungsi keanggotaan dari himpunan *Fuzzy* yang telah ditentukan sebelumnya dapat dilakukan proses perhitungan untuk menentukan seorang individu merupakan klasifikasi apa pada himpunan *Fuzzy* tersebut dan berapa nilai atau derajat keanggotaan individu tersebut dalam himpunan *Fuzzy*. Sebagai contoh untuk individu dengan berat badan 45 merupakan individu dengan berat yang dapat dikatakan RINGAN dan memiliki derajat keanggotaan 1.

2.4.5 Operasi Himpunan *Fuzzy*

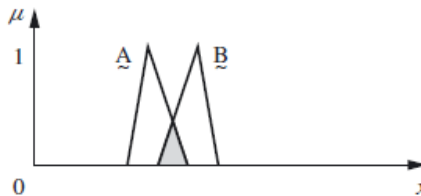
Himpunan *Fuzzy* dapat dioperasikan seperti himpunan konvensional lainnya. Hasil dari operasi himpunan *fuzzy* dapat disebut juga sebagai α -predikat. Himpunan fuzzy memiliki 3 operator :[3]

1. Operator *AND*

Operator *AND* berhubungan dengan operasi irisan himpunan. Hasil dari operator *AND* diperoleh dengan cara mengambil nilai atau derajat keanggotaan terkecil pada himpunan-himpunan yang berhubungan. Nilai keanggotaan operator *AND* untuk himpunan *fuzzy* A dan B dapat dilihat pada persamaan 2.6 dan 2.7.[3]

$$\mu_{A \cap B}(x) = \mu_A \wedge \mu_B \quad (2.6)$$

$$\mu_{A \cap B}(x) = \min(\mu_A [x], \mu_B [y]) \quad (2.7)$$



Gambar 2.8 Irisan *Fuzzy* Set A dan B[3]

Sebagai contoh diketahui,

$$\mu_{\text{BeratBadan_ringan}}(55) = 0,67$$

$$\mu_{\text{BeratBadan_sedang}}(55) = 0,33$$

maka hasil dari operasi AND adalah sebagai berikut

$$\mu_{ringan \cap sedang}(55) = \min(\mu_{ringan}[x] \cdot \mu_{sedang}[x])$$

$$\mu_{ringan \cap sedang}(55) = \min(0,67 ; 0,33)$$

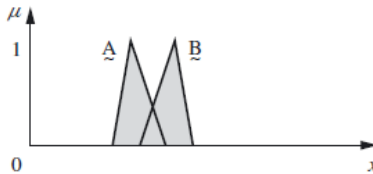
$$\mu_{ringan \cap sedang}(55) = 0,33$$

2. Operator OR

Operator *OR* berhubungan dengan operasi gabungan himpunan. Hasil dari operator *OR* diperoleh dengan cara mengambil nilai atau derajat keanggotaan terbesar pada himpunan-himpunan yang berhubungan. Nilai keanggotaan dari operator *OR* untuk himpunan fuzzy A dan B dapat dilihat pada persamaan 2.8 dan 2.9.[3]

$$\mu_{A \cup B}(x) = \mu_A \vee \mu_B \quad (2.8)$$

$$\mu_{A \cup B}(x) = \max(\mu_A[x] \cdot \mu_B[y]) \quad (2.9)$$



Gambar 2.9 Gabungan Fuzzy Set A dan B[3]

Sebagai contoh diketahui,

$$\mu_{BeratBadan_ringan}(55) = 0,67$$

$$\mu_{BeratBadan_sedang}(55) = 0,33$$

maka hasil dari operasi AND adalah sebagai berikut

$$\mu_{ringan \cap sedang}(55) = \max(\mu_{ringan}[x] \cdot \mu_{sedang}[x])$$

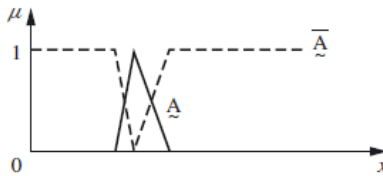
$$\mu_{ringan \cap sedang}(55) = \max(0,67 ; 0,33)$$

$$\mu_{ringan \cap sedang}(55) = 0,67$$

3. Operator *NOT*

Operator *NOT* berhubungan dengan operasi komplement himpunan. Nilai keanggotaan dari operator *NOT* untuk himpunan *fuzzy* A dan B dapat dilihat pada persamaan 2.8.[3]

$$\mu_{\bar{A}}(x) = 1 - \mu_A \quad (2.10)$$



Gambar 2.10 Komplement Fuzzy Set A[3]

Sebagai contoh diketahui,

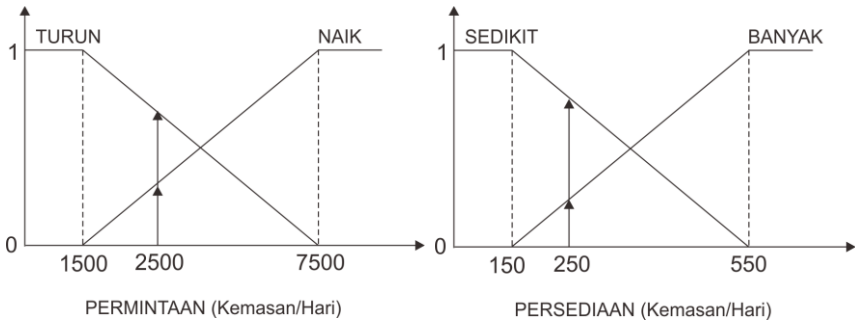
$$\mu_{\text{BeratBadan_ringan}}(55) = 0,67$$

maka hasil dari operasi AND adalah sebagai berikut

$$\begin{aligned} \mu_{\text{ringan}}(55) &= 1 - \mu_A \\ &= 1 - 0,67 \\ &= 0,33 \end{aligned}$$

2.4.6 Fuzzyfikasi

Fuzzyfikasi (tahap pengaburan) merupakan tahap pemetaan nilai masukan berupa nilai tegas (*crisp data*) yang diubah ke dalam bentuk nilai masukan *fuzzy* yang berupa derajat keanggotaan atau tingkat kebenaran.[1] Pada tahap ini nilai tegas (*crisp data*) diambil untuk ditentukan dimana nilai-nilai tersebut menjadi anggota dari setiap himpunan *fuzzy* yang sesuai. Sebagai contoh dalam kasus pada sebuah perusahaan susu kaleng dengan fungsi keanggotaan dan data masukan sebagai berikut,



Gambar 2.11 Fungsi Keanggotaan (Contoh Kasus)

Nilai fungsi keanggotaan permintaan :

$$\mu_{pmtTURUN}(2500) = \frac{b - x}{b - a} = \frac{7500 - 2500}{7500 - 1500} = \frac{5000}{6000} = 0,83$$

$$\mu_{pmtNAIK}(2500) = \frac{x - a}{b - a} = \frac{2500 - 1500}{7500 - 1500} = \frac{1000}{6000} = 0,17$$

Nilai fungsi keanggotaan persediaan :

$$\mu_{psdSEDIKIT}(250) = \frac{b - x}{b - a} = \frac{550 - 250}{550 - 150} = \frac{300}{400} = 0,75$$

$$\mu_{psdBANYAK}(250) = \frac{x - a}{b - a} = \frac{250 - 150}{550 - 150} = \frac{100}{400} = 0,25$$

Jadi, nilai *fuzzy* untuk masukan sebesar 2500 pada fungsi keanggotaan *fuzzy* permintaan turun adalah sebesar 0,83 dan nilai *fuzzy* pada fungsi keanggotaan *fuzzy* permintaan naik adalah sebesar 0,17. Sedangkan nilai *fuzzy* untuk masukan sebesar 250 pada fungsi keanggotaan *fuzzy* persediaan sedikit adalah sebesar 0,75 dan untuk fungsi keanggotaan persediaan banyak adalah sebesar 0,25.

2.4.7 Sistem Inferensi

Tahap kedua pada metode *Fuzzy* yaitu sistem inferensi. Pada tahap ini telah dilakukan pembentukan basis pengetahuan *fuzzy* (*Rule*) yang akan digunakan untuk tahap evaluasi masukan yang telah diubah menjadi derajat keanggotaan *fuzzy* terhadap aturan atau *rule fuzzy* untuk menghasilkan *output* dari tiap-tiap *rule* yang telah dibangun di dalam sistem. *Output* atau keluaran dari tahap ini berupa *output fuzzy*. Sebagai contoh pada kasus perusahaan makanan kaleng yang telah dibahas sebelumnya, diketahui terdapat 4 aturan sebagai berikut :

Rule 1

IF permintaan TURUN and persediaan BANYAK THEN produksi barang BERKURANG

Rule 2

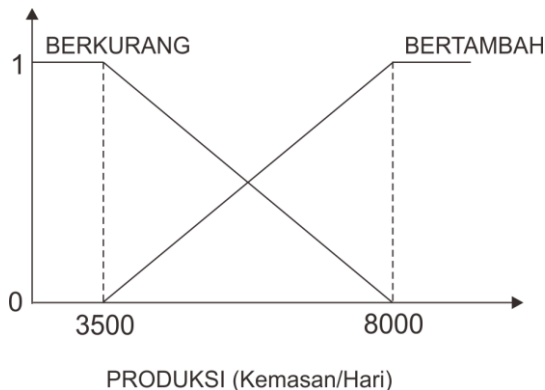
IF permintaan TURUN and persediaan SEDIKIT THEN produksi barang BERKURANG

Rule 3

IF permintaan NAIK and persediaan BANYAK THEN produksi barang BERTAMBAH

Rule 4

IF permintaan NAIK and persediaan SEDIKIT THEN produksi barang BERTAMBAH



Gambar 2.12 Fungsi Keanggotaan 2 (Contoh Kasus)

Hasil dari penarikan kesimpulan pada sistem inferensi,

Rule 1

$$\begin{aligned}
 \alpha - predikat_1 &= \mu_{pmtTURUN} \cap \mu_{psdBANYAK} \\
 &= \min(\mu_{pmtTURUN}[2500] \cap \mu_{psdBANYAK}[250]) \\
 &= \min(0,83; 0,25) \\
 &= 0,25
 \end{aligned}$$

Produksi barang **BERKURANG**,

$$\begin{aligned}
 \frac{8000 - z}{4500} &= 0,25 \\
 8000 - z &= 1125 \\
 z &= 8000 - 1125 \\
 z &= 6875
 \end{aligned}$$

Rule 2

$$\begin{aligned}
 \alpha - predikat_2 &= \mu_{pmtTURUN} \cap \mu_{psdSEDIKIT} \\
 &= \min(\mu_{pmtTURUN}[2500] \cap \mu_{psdSEDIKIT}[250]) \\
 &= \min(0,83; 0,75) \\
 &= 0,75
 \end{aligned}$$

Produksi barang **BERKURANG**,

$$\begin{aligned}
 \frac{8000 - z}{4500} &= 0,75 \\
 8000 - z &= 3375 \\
 z &= 8000 - 3375 \\
 z &= 4625
 \end{aligned}$$

Rule 3

$$\begin{aligned}
\alpha - predikat_3 &= \mu_{pmtNAIK} \cap \mu_{psdBANYAK} \\
&= \min(\mu_{pmtNAIK}[4000] \cap \mu_{psdBANYAK}[300]) \\
&= \min(0,17; 0,25) \\
&= 0,17
\end{aligned}$$

Produksi barang **BERTAMBAH**,

$$\begin{aligned}
\frac{z - 3500}{4500} &= 0,4 \\
z - 3500 &= 1800 \\
z &= 3500 + 1800 \\
z &= 5300
\end{aligned}$$

Rule 4

$$\begin{aligned}
\alpha - predikat_4 &= \mu_{pmtTURUN} \cap \mu_{psdSEDIKIT} \\
&= \min(\mu_{pmtTURUN}[4000] \cap \mu_{psdSEDIKIT}[300]) \\
&= \min(0,83; 0,75) \\
&= 0,75
\end{aligned}$$

Produksi barang **BERTAMBAH**,

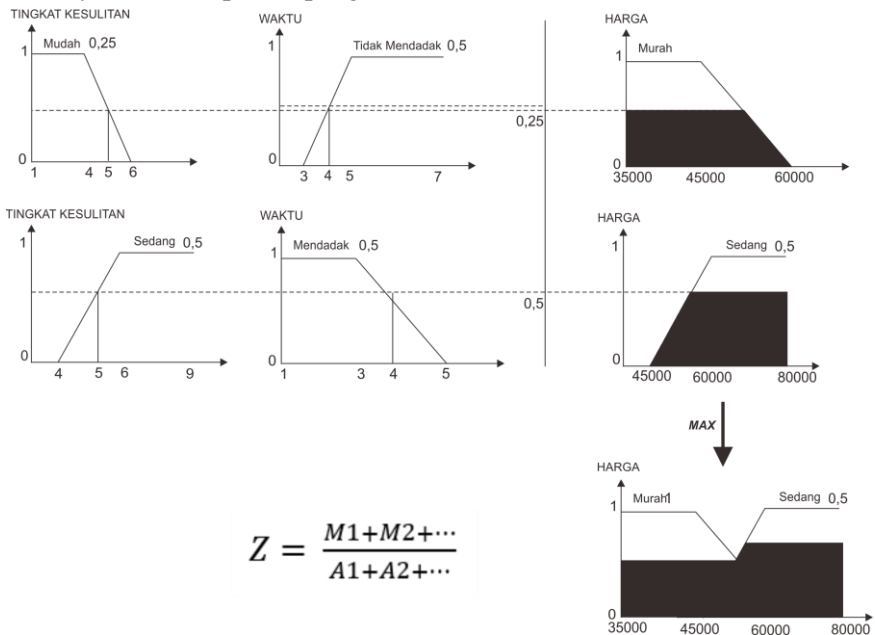
$$\begin{aligned}
\frac{z - 3500}{4500} &= 0,75 \\
z - 3500 &= 3375 \\
z &= 3500 + 3375 \\
z &= 6875
\end{aligned}$$

2.4.8 Defuzzyfikasi

Defuzzyfikasi (tahap penegasan) merupakan tahap dimana dilakukan transformasi hasil dari penarikan kesimpulan pada sistem inferensi menjadi *output* berupa nilai *crisp* atau nilai tegas. Terdapat 7 metode yang dapat digunakan dalam proses defuzzyfikasi antara lain yaitu *Height Method*, *Centroid (Center of Gravity) Method*, *Weighted Average Method*, *Mean-Max*, *Center of Sums*, *Center of Largest Area*, *First (or last) of Maxima*. [4]

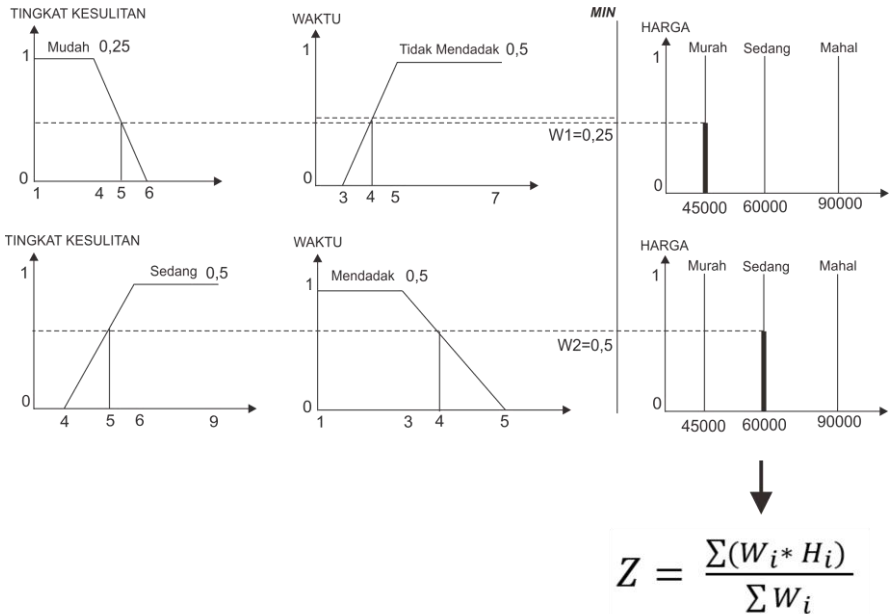
2.4.9 Metode Tsukamoto

Terdapat beberapa metode yang dapat digunakan pada sistem inferensi fuzzy, antara lain yaitu metode mamdani, sugeno dan tsukamoto. Perbedaan pada ketiga metode tersebut dapat diamati pada beberapa ilustrasi berikut. Seperti yang dapat kita lihat pada **Gambar 2.13**, pada metode Mamdani menggunakan 2 operator berupa operator MIN dan MAX. Setelah proses evaluasi *rule* akan dilakukan proses agregasi atau penggabungan hasil dari evaluasi untuk dilakukan proses defuzzyfikasi atau proses pengembalian nilai.



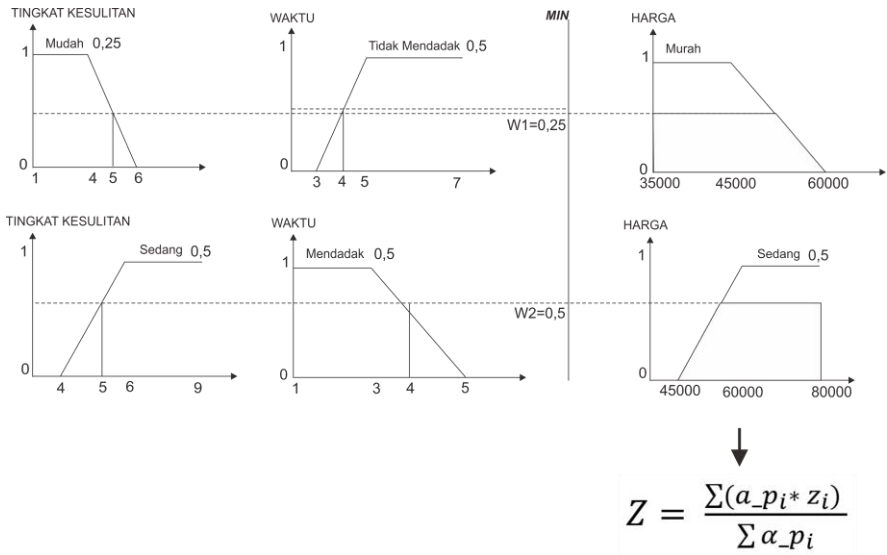
Gambar 2.13 Ilustrasi Metode Mamdani

Berbeda dengan ilustrasi metode Mamdani, pada **Gambar 2.14** dapat diamati bahwa metode Sugeno hanya menggunakan 1 operator himpunan berupa operator MIN yang mempermudah dalam proses evaluasi *rule*. Selain itu hasil dari evaluasi *rule* pada metode ini berupa sebuah persamaan matematis pada masing-masing domain.



Gambar 2.14 Ilustrasi Metode Sugeno

Tidak jauh berbeda dengan metode sugeno, pada **Gambar 2.15** dapat diamati bahwa metode Tsukamoto hanya menggunakan 1 operator himpunan yaitu MIN. Selain itu *output* fuzzy pada metode Tsukamoto memungkinkan keluaran berbentuk rentan keluaran yang dapat direpresentasikan dalam sebuah fungsi keanggotaan.



Gambar 2.15 Ilustrasi Metode Tsukamoto

Metode Tsukamoto dipilih pada proses perhitungan harga akhir ini karena cocok dengan permasalahan yang ada dan mudah untuk diimplementasikan pada proses perhitungan yang memiliki keluaran berupa rentan harga. Pada proses sistem inferensi metode Tsukamoto juga hanya menggunakan 1 operator yaitu operator AND (MIN) dengan cara mengambil nilai minimum pada himpunan-himpunan yang berkaitan. Proses defuzzyfikasi pada metode ini menggunakan *Weighted Average Method* atau metode rata-rata terbobot untuk mengurangi penggunaan waktu pada proses defuzzyfikasi, rumus yang digunakan pada metode tersebut adalah sebagai berikut :[1]

$$Z = \frac{\sum(a_{p_i} * z_i)}{\sum \alpha_{p_i}} \quad (2.11)$$

Keterangan :

Z = Hasil *Defuzzyfication* metode rata-rata terbobot

a_{p_i} = Nilai minimum derajat keanggotaan

z_i = Nilai *crisp* hasil inferensi rule

i = Jumlah aturan *fuzzy*

Sebagai contoh pada kasus perusahaan makanan kaleng yang sebelumnya telah dibahas, dapat diketahui hasil defuzzyfikasi dengan metode *Weighted Average Method* adalah sebagai berikut,

$$\begin{aligned}
 Z &= \frac{\text{apred}_1 * z_1 + \text{apred}_2 * z_2 + \text{apred}_3 * z_3 + \text{apred}_4 * z_4}{\text{apred}_1 + \text{apred}_2 + \text{apred}_3 + \text{apred}_4} \\
 Z &= \frac{0,25 * 6875 + 0,75 * 4625 + 0,17 * 5300 + 0,75 * 6875}{0,25 + 0,75 + 0,17 + 0,75} \\
 Z &= \frac{11244,75}{1,92} \\
 Z &= 586
 \end{aligned}$$

2.5 Java

Java adalah bahasa pemrograman berorientasi objek murni yang berisi beberapa bagian program berupa beberapa kelas.[6] Bahasa pemrograman Java dapat dijalankan pada berbagai komputer maupun telepon genggam pada beberapa *platform* sistem operasi yang berbeda. Bahasa ini banyak mengadopsi sintaksis pada C dan C++ yang disederhanakan. Kelebihan dari bahasa pemrograman ini yaitu *multiplatform*, OOP (*Object Oriented Programming* – Pemrogram Berorientasi Objek), memiliki perpustakaan (*library*) kelas yang lengkap, bergaya C++ dan memiliki fasilitas pengaturan penggunaan memori secara otomatis.[6] Kekurangan dari bahasa pemrograman ini sendiri yaitu masih ada beberapa hal yang tidak kompatibel antara *platform* satu dengan yang lainnya, mudah didekompilasi dan penggunaan memori yang jauh lebih besar dari sebelumnya.[6]

2.6 NetBeans IDE

NetBeans IDE merupakan salah satu alat bantu dalam pengembangan perangkat lunak yang dituliskan dalam bahasa pemrograman Java. Aplikasi berbasis *NetBeans Platform* dapat dikembangkan kembali oleh pengembang pihak ketiga. Pada umumnya aplikasi yang dikembangkan menggunakan bahasa Java, namun *NetBeans IDE* juga mendukung bahasa pemrograman lainnya seperti PHP, C/C++ dan HTML5.[7]

NetBeans IDE mendukung pengembangan aplikasi *desktop* profesional, aplikasi perangkat bergerak (*mobile*) dan aplikasi

enterprise. IDE juga menyediakan *wizards* dan *templates* untuk membuat aplikasi Java EE, Java SE dan Java ME.[7] Selain untuk membangun sebuah aplikasi, IDE juga dapat digunakan untuk *refactoring* kode program yang sudah dibuat. Banyak sekali variasi dari teknologi dan *frameworks* yang dapat digunakan pada NetBeans IDE. *Tools* ini juga memiliki fitur *debugging* dan *profiling* seperti kasus-kasus *deadlocks* maupun kebocoran memori. Ketika aplikasi sudah selesai dibangun terdapat fitur *testing* dan analisa kode dimana IDE menyediakan beberapa *tools* seperti JUnit dan TestNG sebagai penganalisa kode program yang ada.[7]

2.7 Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu – Integrated Development Environment (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA.[8] Setiap proyek yang dibangun pada Android Studio memiliki beberapa modul dengan file kode sumber dan file lainnya yang mencakup modul aplikasi Android, modul pustaka dan modul *Google App Engine*. Pada masing-masing modul aplikasi terdiri atas 3 folder yaitu folder manifest yang berisi file *AndroidManifest.xml*, folder java yang berisi kode sumber Java termasuk kode pengujian JUnit dan folder res dimana berisi semua sumber daya bukan kode seperti file XML yang berisi tata letak, string UI dan gambar bitmap.[8]

Android Studio lebih banyak dipilih karena memiliki banyak fitur dan *library* siap pakai yang memudahkan *programmer* level dasar maupun *profesional*. Beberapa fitur unggulan Android Studio yaitu sistem versi yang berbasis Gradle yang fleksibel, emulator yang cepat dan kaya fitur, *Instant Run* untuk perubahan pada aplikasi tanpa membuat APK baru, *auto completion* pada penulisan kode program, sistem *build* yang handal dan fleksibel karena tidak perlu lagi *build* aplikasi untuk memperoleh *file* APK dan fitur dimana pengguna dapat membuat aplikasi untuk semua perangkat berbasis Android seperti *Smartwatch*, *Tablet*, *Android TV*, dan *Android Auto*.[8]

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Bab analisis dan perancangan berisi analisis kebutuhan dan perancangan aplikasi yang akan dibangun. Tahap analisis membahas mengenai analisis kebutuhan yang menjadi dasar dari tahap perancangan. Kebutuhan yang menjadi dasar dari tahap perancangan berupa data dimana data tersebut akan digunakan sebagai masukan dan keluaran pada proses perhitungan harga tutor. Pada tahap perancangan sistem, hasil analisis data tersebut akan digunakan dalam serangkaian proses. Proses tersebut merupakan beberapa tahapan yang ada dalam sistem, yang akan melakukan perhitungan terhadap data-data tersebut dengan menggunakan metode *Fuzzy Tsukamoto*.

3.1 Tahap Analisis

Tahap analisis mendefinisikan kebutuhan yang akan dipenuhi dalam pembangunan aplikasi "*Finding Tutor*" khususnya pada perhitungan harga tutor. Selain itu dijelaskan pula alasan pengerjaan masing-masing tahap pada tugas akhir ini.

3.1.1 Deskripsi Umum Sistem

Pada tugas akhir ini dibangun aplikasi "*Finding Tutor*" yang dapat dimanfaatkan untuk melakukan pencarian tutor sesuai dengan kebutuhan dan waktu yang diinginkan konsumen serta harga yang bersaing. Data masukan yang digunakan adalah data tingkat kesulitan materi, waktu jasa digunakan serta jarak antara tutor dan murid (konsumen). Data keluaran dari aplikasi merupakan harga yang merepresentasikan hasil dari perhitungan sistem dengan metode *Fuzzy Tsukamoto* yang akan diolah kembali dengan masukan konsumen berupa durasi dan ketentuan harga dari zona-zona daerah yang telah ditetapkan sebelumnya.

Aplikasi ini diharapkan dapat mempermudah masyarakat untuk mendapatkan guru les privat (tutor) yang sesuai dengan kebutuhan dan waktu yang diinginkan konsumen. Selain itu harga dari hasil perhitungan dengan menggunakan metode *Fuzzy*

Tsukamoto ini diharapkan dapat dimanfaatkan sebagai daya tarik pengguna aplikasi “*Finding-Tutor*” dimana harga yang dihasilkan merupakan harga yang bersaing dan menguntungkan untuk kedua belah pihak baik penyedia jasa (tutor) maupun konsumen (murid). Skema proses bisnis pada aplikasi dapat dilihat pada **Gambar 3.1**.

3.1.2 Data

Pada sub-bab ini akan dijelaskan mengenai data yang digunakan sebagai masukan serta parameter dalam proses perhitungan harga dengan metode *Fuzzy Tsukamoto* yang selanjutnya akan diolah dengan parameter lainnya berupa durasi dan zona-zona daerah yang telah ditetapkan sebelumnya.

3.1.2.1 Data Masukan

Data masukan merupakan data yang dimasukkan oleh *user* yang akan digunakan sebagai masukan dari sistem maupun parameter dalam proses perhitungan harga tutor. Data masukan yang akan digunakan dalam perhitungan harga tutor menggunakan metode *Fuzzy Tsukamoto* adalah tingkat kesulitan, waktu dan jarak antara tutor dengan konsumen (murid). Data masukan tersebut merupakan hal-hal yang mempengaruhi harga dalam penentuan harga tutor yang didapatkan dari hasil kuisioner di lapangan,

Data masukan tingkat kesulitan berupa nilai 1-13 dimana nilai tersebut mendeskripsikan tingkat kesulitan tiap-tiap kelas yang ada pada tingkat pendidikan. Keterangan data masukan berupa tingkat kesulitan ditunjukkan pada **Tabel 3.1**.

Data masukan waktu berupa nilai 0 - 14 dimana nilai tersebut mendeskripsikan seberapa jauh konsumen memesan *tutor*. Rentan pemesanan tutor dibatasi maksimal 2 minggu sebelum jadwal mengajar. Nilai masukan waktu didapatkan dari selisih masukan konsumen berupa tanggal transaksi tutor mengajar (jadwal mengajar) dengan tanggal sistem dimana transaksi dilakukan oleh konsumen (waktu pemesanan).

Data masukan jarak berupa nilai 0 – 35.000 m dimana nilai tersebut mendeskripsikan jarak antara alamat tutor dengan alamat konsumen atau pemesanan tutor. Selisih jarak tersebut didapatkan

dari selisih jarak alamat tutor yang telah terdaftar dengan data masukan alamat pada formulir pemesanan.

Tabel 3.1 Masukan Tingkat Kesulitan

Tingkat Kesulitan	Keterangan
1	Kelas 1 SD
2	Kelas 2 SD
3	Kelas 3 SD
4	Kelas 4 SD
5	Kelas 5 SD
6	Kelas 6 SD
7	Kelas 1 SMP
8	Kelas 2 SMP
9	Kelas 3 SMP
10	Kelas 1 SMA
11	Kelas 2 SMA
12	Kelas 3 SMA
13	Umum

3.1.2.2 Data Keluaran

Data keluaran berupa hasil representasi dari proses perhitungan yang dilakukan pada implementasi metode *Fuzzy Tsukamoto*. Data keluaran berupa harga tersebut dibangun dalam fungsi keanggotaan untuk menunjukan rentan harga yang telah didapatkan dari hasil kuisioner dilapangan. Data keluaran yang dihasilkan dari perhitungan kemudian akan diolah kembali dengan data lainnya berupa durasi dan ketentuan zona daerah. Keterangan data keluaran berupa harga ditunjukkan pada **Tabel 3.2**.

Tabel 3.2 Kategori Harga Durasi 90 Menit

Harga	Keterangan
Rp 35.000 – Rp 60.000	Murah
Rp 45.000 – Rp 80.000	Sedang
Rp 80.000 – Rp 100.000	Mahal

3.1.2.3 Data Lainnya

Data lainnya merupakan data yang akan digunakan dalam perhitungan harga akhir tutor. Data lainnya yang akan digunakan yaitu data durasi yang didapatkan dari data masukan *user* serta data zona-zona daerah (**Tabel 3.3**) yang telah ditentukan sebelumnya.

3.1.3 Analisis Permasalahan

Proses yang ada pada sistem akan mengolah data yang ada untuk mendapatkan harga akhir tutor yang akan digunakan pada aplikasi dengan menggunakan metode *Fuzzy Tsukamoto*. Proses perhitungan tersebut akan menghasilkan harga mentah yang harus diolah kembali karena kendala jarak pada daerah-daerah tertentu yang tidak terjangkau oleh tutor.

Karena luasnya daerah jangkauan aplikasi yaitu pada daerah Surabaya, maka dibutuhkan pembagian daerah yang ada di kota Surabaya menjadi 3 zona untuk menentukan tambahan biaya transportasi yang akan diberikan pada tutor. Nama zona beserta ketentuan yang telah ditetapkan dapat dilihat pada **Tabel 3.3**.

Tabel 3.3 Daftar Zona Daerah

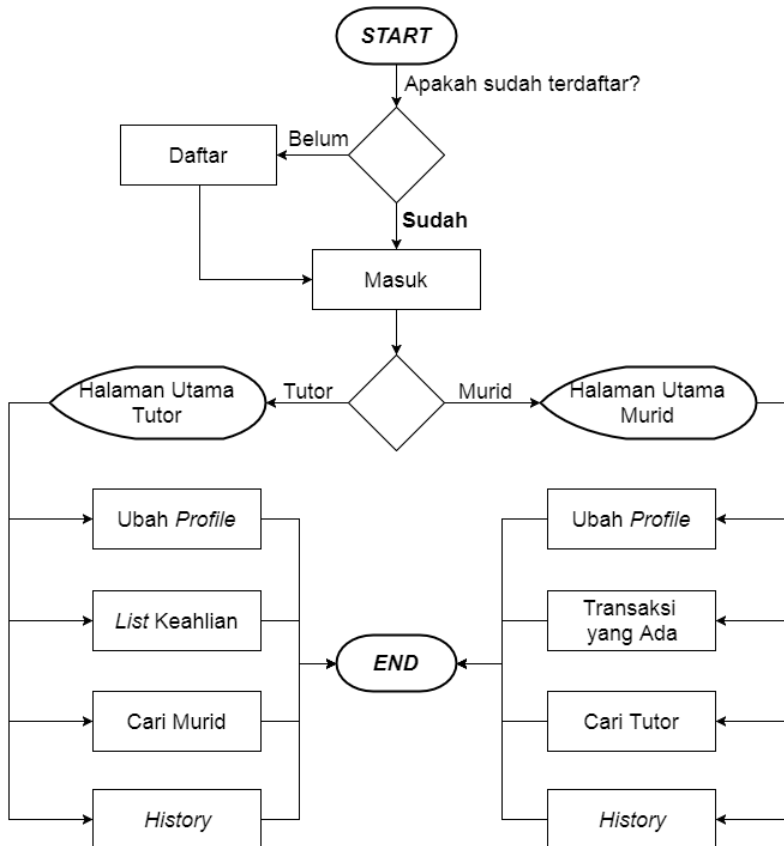
	Deskripsi	Ketentuan
Zona 1	$5 \text{ km} < \text{Jarak} \leq 10 \text{ km}$	Harga + Rp 5.000
Zona 2	$10,1 \text{ m} < \text{Jarak} \leq 15 \text{ km}$	Harga + Rp Rp 10.000
Zona 3	$\text{Jarak} > 15.001 \text{ m}$	Harga + Rp Rp 20.000

3.2 Tahap Perancangan Sistem

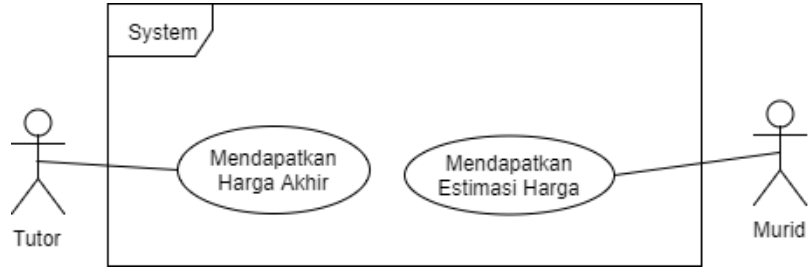
Tahap perancangan dilakukan untuk merancang proses secara keseluruhan berdasarkan fungsionalitas dan kebutuhan dari perhitungan harga yang ada pada aplikasi “*Finding Tutor*”.

3.2.1 Desain Umum Sistem

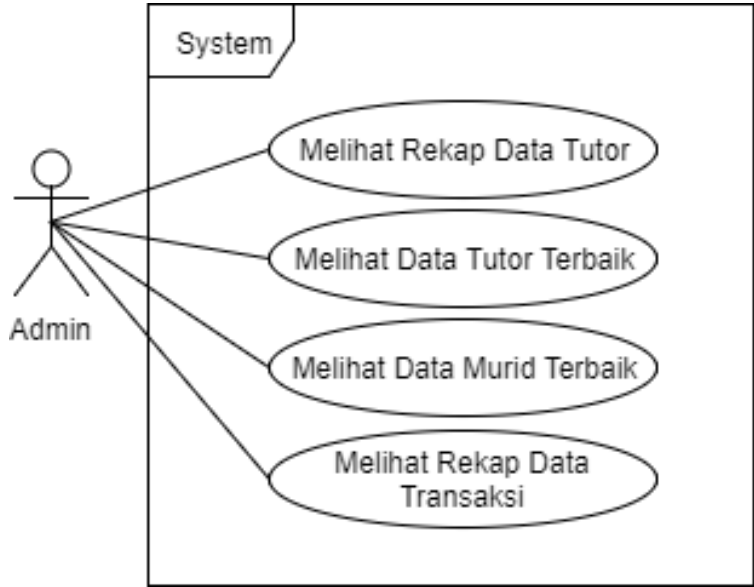
Pada sub-bab ini akan dijelaskan bagaimana sistem berjalan secara keseluruhan. Diagram alir sistem secara keseluruhan dapat dilihat pada **Gambar 3.1**,



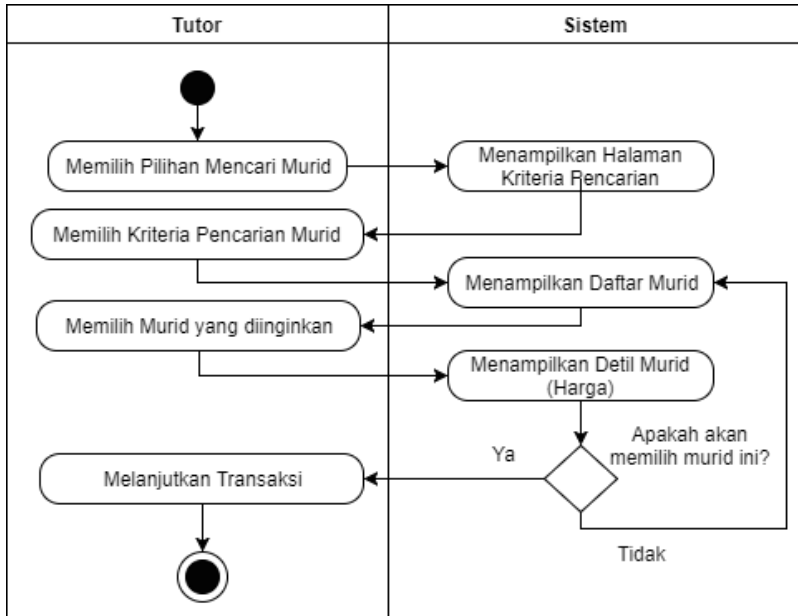
Gambar 3.1 Diagram Alir Sistem



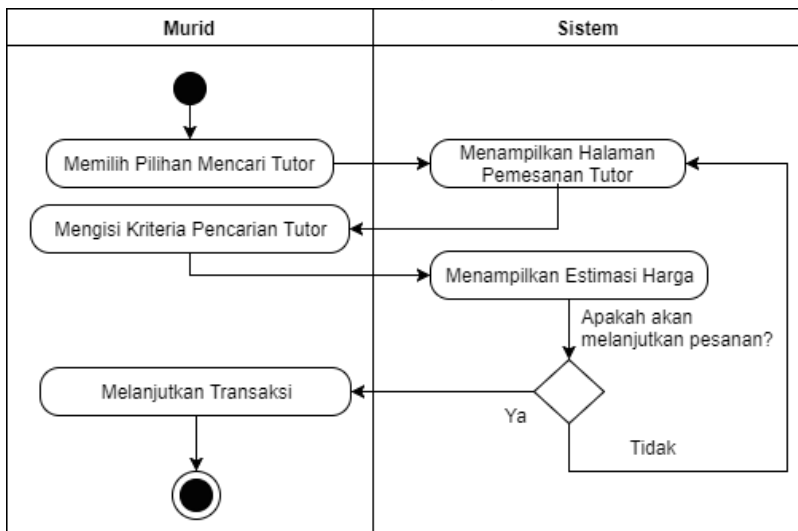
Gambar 3.2 Use Case Diagram Aplikasi *Finding Tutor*



Gambar 3.3 Use Case Diagram *Website*



Gambar 3.4 Activity Diagram Tutor



Gambar 3.5 Activity Diagram Murid

3.2.2 Perancangan Data

Pada proses perhitungan harga tutor dengan metode *Fuzzy Tsukamoto* membutuhkan masukan berupa 3 variabel yang telah dibangun dalam fungsi keanggotaan yang dibutuhkan metode *Fuzzy*.

3.2.2.1 Data Masukan

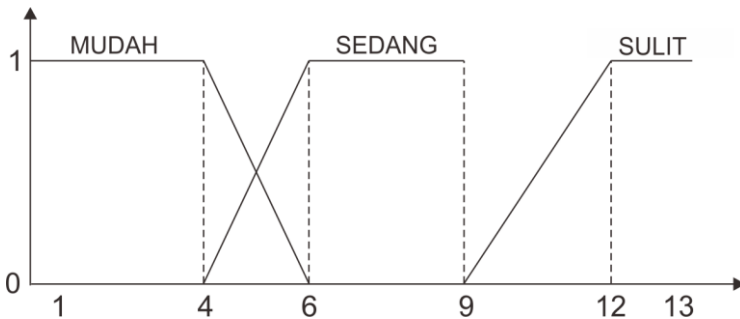
Pada proses perhitungan harga tutor membutuhkan variabel yang digunakan dalam perhitungan berupa data masukan yang diberikan *user* ke dalam sistem berupa tingkat kesulitan, waktu dan jarak. Ketiga variabel tersebut dibangun dalam fungsi keanggotaan untuk melanjutkan proses perhitungan. Fungsi keanggotaan yang telah dibangun berdasarkan variabel-variabel yang telah ditentukan dapat ditunjukkan pada **Gambar 3.2 – Gambar 3.4**.

Fungsi keanggotaan Tingkat Kesulitan :

$$\mu_{Tingkat_Kesulitan_Mudah}[x] = \begin{cases} 1, & x \geq 1 \text{ dan } x \leq 4 \\ \frac{6-x}{4}, & 4 < x < 6 \\ 0, & x < 1 \text{ dan } x \geq 6 \end{cases} \quad (3.1)$$

$$\mu_{Tingkat_Kesulitan_Sedang}[x] = \begin{cases} 1, & 6 < x \leq 9 \\ \frac{x-4}{2}, & 4 < x \leq 6 \\ 0, & x \leq 4 \text{ dan } x > 9 \end{cases} \quad (3.2)$$

$$\mu_{Tingkat_Kesulitan_Sulit}[x] = \begin{cases} 1, & 12 < x \leq 13 \\ \frac{x-9}{3}, & 9 < x \leq 12 \\ 0, & x \leq 9 \text{ dan } x > 13 \end{cases} \quad (3.3)$$



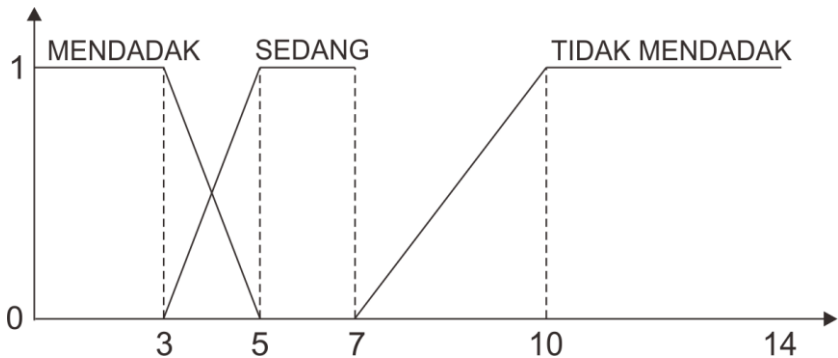
Gambar 3.6 Fungsi Keanggotaan Tingkat Kesulitan

Fungsi keanggotaan Waktu :

$$\mu_{Waktu_Mendadak}[x] = \begin{cases} 1, & x \geq 0 \text{ dan } x \leq 3 \\ \frac{5-x}{2}, & 3 < x < 5 \\ 0, & x < 1 \text{ dan } x \geq 5 \end{cases} \quad (3.4)$$

$$\mu_{Waktu_Sedang}[x] = \begin{cases} 1, & 5 < x \leq 7 \\ \frac{x-3}{2}, & 3 < x \leq 5 \\ 0, & x \leq 3 \text{ dan } x > 7 \end{cases} \quad (3.5)$$

$$\mu_{Waktu_Tidak_Mendadak}[x] = \begin{cases} 1, & 10 \leq x \leq 14 \\ \frac{x-7}{3}, & 7 < x < 10 \\ 0, & x \leq 7 \text{ dan } x > 14 \end{cases} \quad (3.6)$$



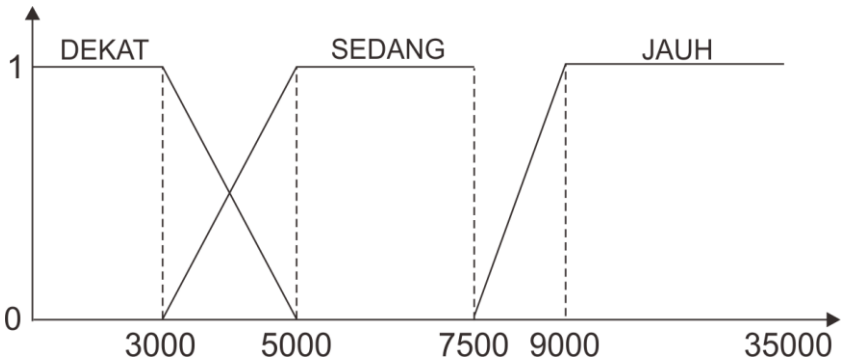
Gambar 3.7 Fungsi Keanggotaan Waktu

Fungsi keanggotaan Jarak :

$$\mu_{Jarak_Dekat}[x] = \begin{cases} 1, & x \geq 0 \text{ dan } x \leq 3000 \\ \frac{5000-x}{2000}, & 3000 < x < 5000 \\ 0, & x \geq 5000 \end{cases} \quad (3.7)$$

$$\mu_{Jarak_Sedang}[x] = \begin{cases} 1, & 5000 < x \leq 7500 \\ \frac{x-3000}{2000}, & 3000 < x \leq 5000 \\ 0, & x \leq 3000 \text{ dan } x > 7500 \end{cases} \quad (3.8)$$

$$\mu_{Jarak_Jauh}[x] = \begin{cases} 1, & 9000 < x \leq 35000 \\ \frac{x-7500}{1500}, & 7500 < x \leq 9000 \\ 0, & x \leq 7500 \text{ dan } x > 35000 \end{cases} \quad (3.9)$$



Gambar 3.8 Fungsi Keanggotaan Jarak

3.2.2.2 Data Keluaran

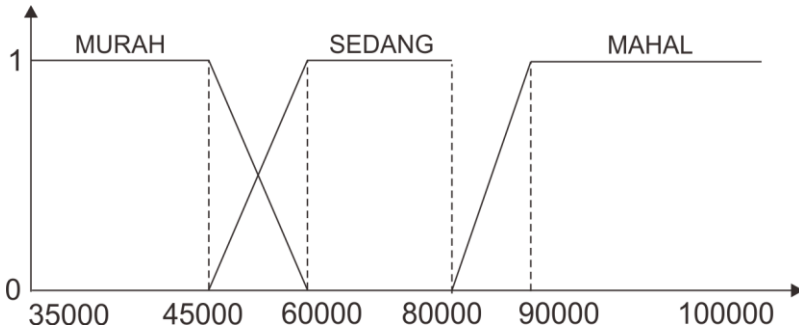
Hasil dari perhitungan harga tutor pada metode *Fuzzy Tsukamoto* berupa harga dimana data keluaran tersebut dibangun dalam sebuah fungsi keanggotaan pada **Gambar 3.5**.

Fungsi keanggotaan Jarak :

$$\mu_{\text{Harga_Murah}}[x] = \begin{cases} 1, & x \geq 35000 \text{ dan } x \leq 45000 \\ \frac{60000-x}{15000}, & 45000 < x < 60000 \\ 0, & x < 35000 \text{ dan } x > 60000 \end{cases} \quad (3.10)$$

$$\mu_{\text{Harga_Sedang}}[x] = \begin{cases} 1, & x \geq 60000 \text{ dan } x \leq 80000 \\ \frac{x-45000}{15000}, & 45000 < x < 60000 \\ 0, & x \leq 45000 \text{ dan } x > 80000 \end{cases} \quad (3.11)$$

$$\mu_{\text{Harga_Mahal}}[x] = \begin{cases} 0, & \leq 80000 \text{ dan } x > 100000 \\ \frac{x-80000}{10000}, & 80000 < x < 90000 \\ 1, & x \geq 90000 \text{ dan } x \leq 100000 \end{cases} \quad (3.12)$$



Gambar 3.9 Fungsi Keanggotaan Harga

3.2.3 Perancangan Proses

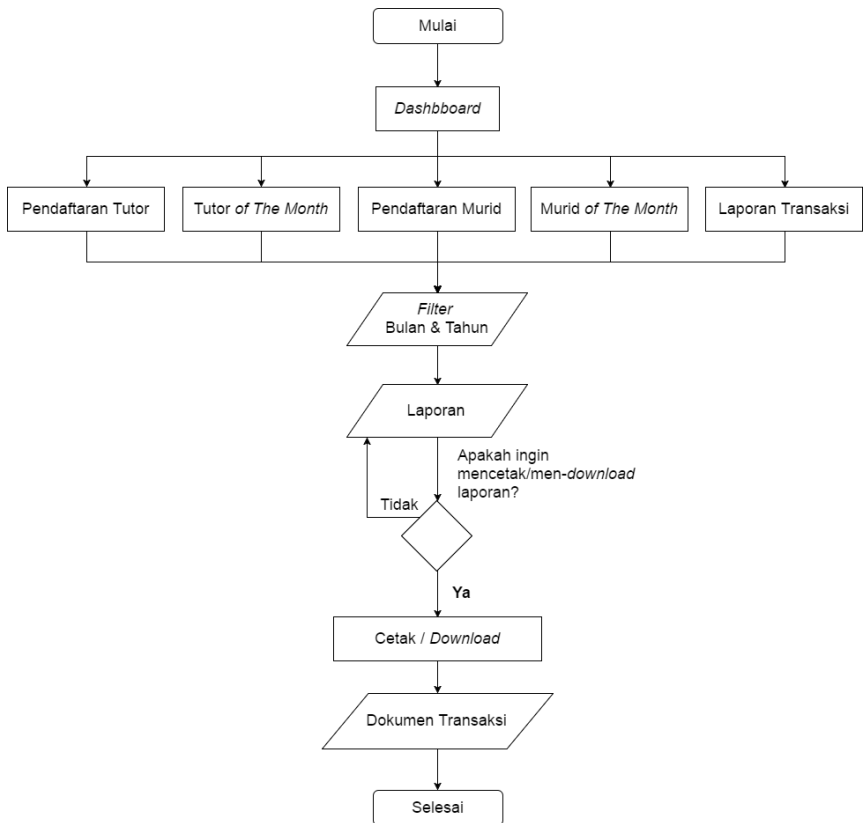
Perancangan proses dilakukan untuk menggambarkan setiap proses yang akan dilakukan pada sistem dari sisi admin, tutor maupun murid. Diagram alir tersebut akan ditunjukkan pada **Gambar 3.6- Gambar 3.8**.

3.2.3.1 Proses pada aktor *Admin*

Admin merupakan salah satu aktor pada proses bisnis yang berada diluar aplikasi “*Finding Tutor*”. *Admin* memiliki akses pada *website* yang telah terintegrasi pada aplikasi untuk menunjukkan perkembangan statistika proses bisnis pada aplikasi “*Finding Tutor*”. *Website* tersebut memiliki beberapa fitur utama dimana *Admin* dapat mengetahui jumlah pendaftar tutor maupun murid, jumlah transaksi dan jumlah pemasukan hari ini secara *up to date*.

Selain fitur utama tersebut, terdapat fitur lain berupa laporan pendaftaran tutor dan murid yang dapat dilihat berdasarkan bulan dan tahun yang diinginkan. Fitur tersebut juga dapat digunakan pada laporan transaksi yang diinginkan. *Admin* juga dapat melihat daftar tutor maupun murid yang masuk ke dalam kategori *Tutor/Murid of The Month*. Kategori *Tutor/Murid of The Month* adalah daftar tutor dan murid terbaik dimana kategori tersebut ditentukan oleh banyaknya transaksi yang telah diambil atau dibuat oleh tutor maupun murid. Daftar ini juga dapat dilihat dengan fitur waktu berupa bulan dan tahun yang diinginkan. Semua data laporan yang ada pada proses bisnis “*Finding Tutor*” dapat dicetak maupun di-*download* dalam bentuk CSV maupun *pdf*.

Untuk memudahkan *Admin* dalam mengamati banyaknya transaksi yang telah dilakukan pada aplikasi, *admin* dapat dengan mudah mengamati hal tersebut melalui laporan yang disediakan dalam bentuk grafik per bulan dan per-tahun. Grafik tersebut dapat di-*download* dalam bentuk gambar maupun *pdf*. Tujuan dari semua fitur yang telah dibangun pada *website* ini adalah untuk memudahkan pihak *developer* maupun manajemen untuk memantau proses bisnis yang ada pada aplikasi “*Finding Tutor*”. Diagram alir proses *admin* dapat dilihat pada **Gambar 3.6**.



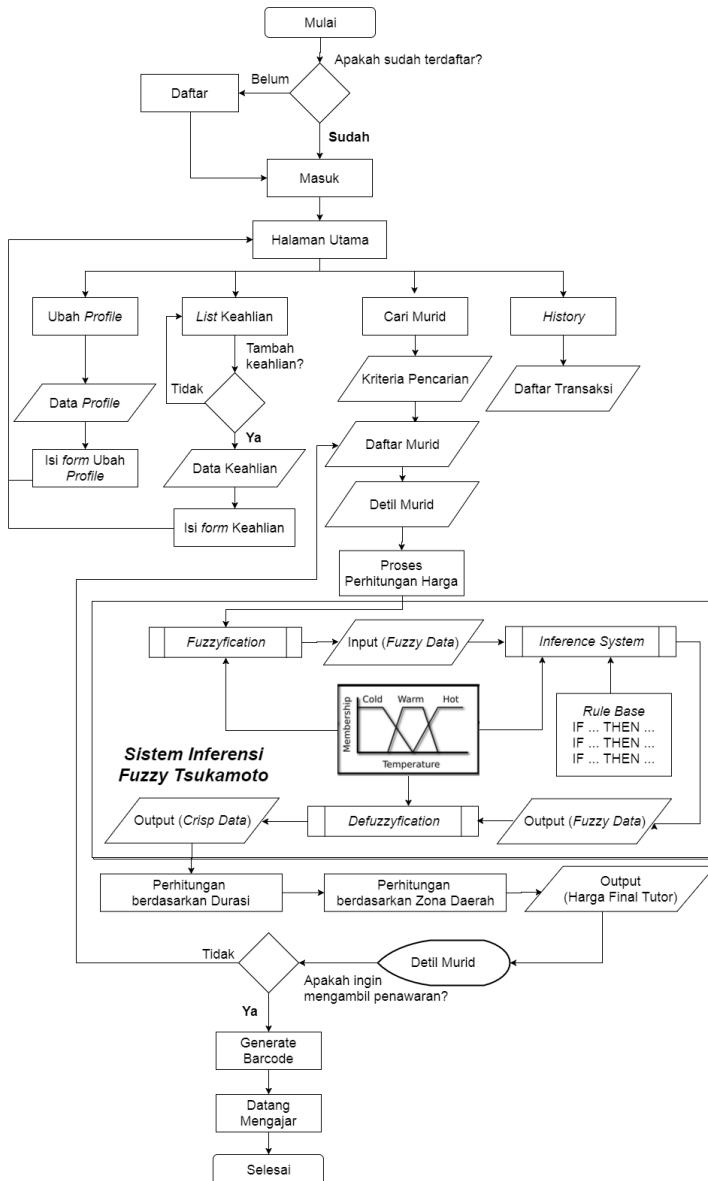
Gambar 3.10 Diagram Alir Admin

3.2.3.2 Proses pada aktor Tutor

Tutor merupakan aktor pada aplikasi “*Finding Tutor*” yang berperan sebagai penyedia jasa. Pada aplikasi “*Finding Tutor*” pengguna diharuskan mendaftarkan diri sebagai tutor sebelum masuk ke dalam proses bisnis yang ada pada aplikasi. Tutor yang telah terdaftar dapat *log-in* dengan username dan password yang telah dibuat sebelumnya.

Pada aplikasi “*Finding Tutor*” untuk tutor, terdapat fitur utama berupa pencarian murid. Pada fitur ini tutor dapat mencari murid dengan kriteria pencarian yang diinginkan. Kriteria pencarian dapat berupa pencarian murid berdasarkan jarak terdekat, pelajaran/keahlian, ketersediaan hari, jenis kelamin, kelas, usia dan tanpa kriteria atau menampilkan semua transaksi murid yang belum diambil oleh tutor. Pada fitur ini juga akan dilakukan perhitungan harga dengan metode *Fuzzy Tsukamoto* dengan masukan berupa tingkat kesulitan, waktu dan jarak antara murid dan tutor yang sedang melakukan proses pencarian murid. Setelah tutor mengambil transaksi murid yang diinginkan, tutor dapat melakukan transaksi tersebut pada waktu yang telah ditentukan murid sebelumnya. Ketika hari tersebut telah datang perhitungan waktu untuk *scan barcode* akan dimulai sesuai dengan waktu yang telah ditentukan murid pada saat proses pemesanan. Tutor diharuskan untuk menunjukkan *barcode* kepada siswa agar di *scan* untuk memastikan bahwa tutor yang datang adalah tutor dari aplikasi “*Finding Tutor*” yang telah mengambil transaksi murid tersebut. Ketika transaksi telah selesai, *barcode* juga harus *discan* kembali untuk menandakan bahwa transaksi antara tutor dan murid tersebut telah selesai.

Sebagai fitur pendukung, pada aplikasi ini juga terdapat fitur history dimana fitur tersebut akan mencatat seluruh transaksi yang telah berhasil dilakukan tutor tersebut selama ini. Fitur pendukung lain yang disediakan berupa daftar keahlian. Pada fitur ini tutor dapat menambahkan keahlian-keahlian yang dimiliki sebagai bahan promosi diri kepada murid. Tutor juga memiliki fitur berupa ubah *profile* yang dapat digunakan untuk menambahkan atau mengubah data diri tutor yang telah diisikan sebelumnya. Diagram alir proses tutor dapat dilihat pada **Gambar 3.7**.



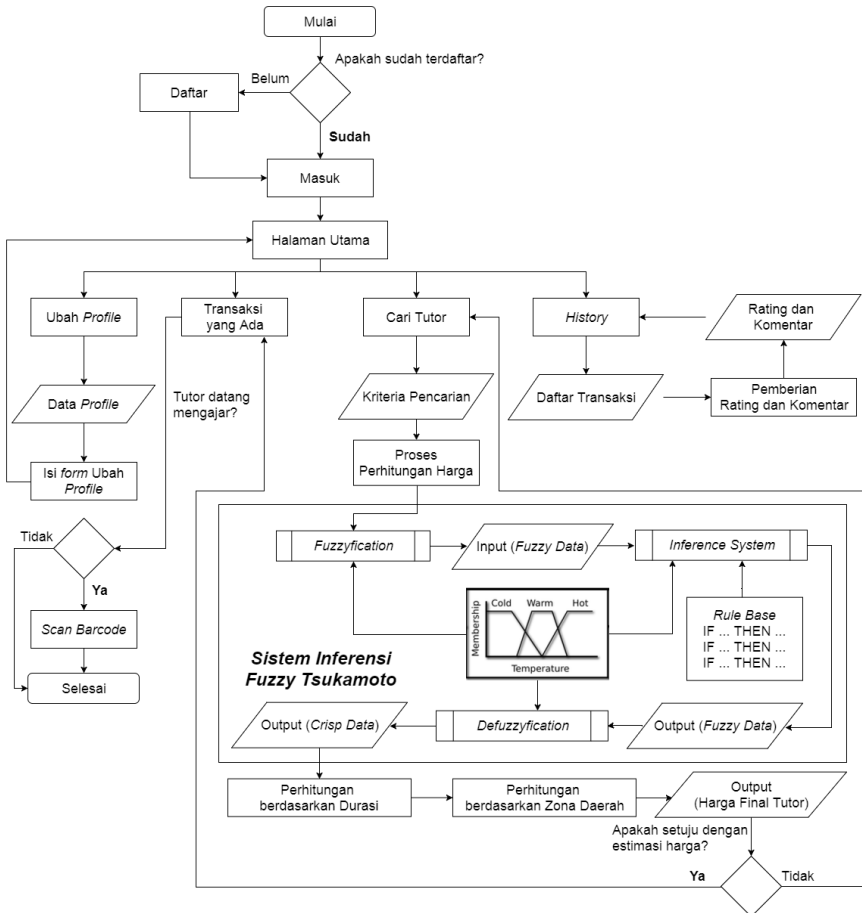
Gambar 3.11 Diagram Alir Tutor

3.2.3.3 Proses pada aktor Murid

Murid merupakan aktor pada aplikasi “*Finding Tutor*” yang berperan sebagai pengguna jasa. Pada aplikasi “*Finding Tutor*” pengguna diharuskan mendaftarkan diri sebagai murid sebelum masuk ke dalam proses bisnis yang ada pada aplikasi. Murid yang telah terdaftar dapat *log-in* dengan username dan password yang telah dibuat sebelumnya.

Pada aplikasi “*Finding Tutor*” untuk murid, terdapat fitur utama berupa pencarian tutor. Pada fitur ini murid dapat mencari tutor dengan kriteria pencarian yang diinginkan dan dibutuhkan. Kriteria pencarian tutor yang dapat dimasukkan antara lain berupa jenis kelamin tutor dan usia tutor yang diinginkan. Pada fitur ini akan dilakukan proses perhitungan estimasi harga dengan metode *Fuzzy Tsukamoto* dengan masukan berupa tingkat kesulitan dan waktu penggunaan jasa. Jika murid telah setuju dengan estimasi harga yang ditawarkan, transaksi akan dilanjutkan dan masuk dalam daftar transaksi yang sedang berjalan. Fitur transaksi yang sedang berjalan merupakan salah satu fitur yang dapat memudahkan murid untuk mengetahui transaksi yang sedang dilakukan dan data dari tutor yang mengambil transaksi tersebut. Ketika transaksi akan dilakukan pada waktu yang telah dicantumkan sebelumnya, murid harus melakukan *scan barcode* yang ada pada tutor untuk memastikan bahwa tutor tersebut adalah tutor yang sesuai dengan data transaksi yang ada. Perhitungan durasi transaksi akan dimulai setelah *scan barcode* dilakukan. Setelah transaksi selesai, siswa juga diharuskan untuk melakukan *scan barcode*

Sebagai fitur pendukung juga terdapat fitur *History*. Fitur ini akan mencatat seluruh transaksi yang telah berhasil dilakukan oleh murid. Selain itu juga terdapat fitur ubah *profile* dimana murid dapat mengubah data diri yang sebelumnya telah dimasukkan.



Gambar 3.12 Diagram Alir Murid

3.2.3.4 Proses Fuzzyfikasi

Proses fuzzyfikasi merupakan tahap pertama dimana proses perhitungan harga dilakukan. Masukan berupa tingkat kesulitan, waktu dan jarak antara murid dan tutor akan diolah menjadi masukan *fuzzy* untuk dapat dilanjutkan ke dalam sistem selanjutnya. Masukan tersebut akan dipetakan ke dalam fungsi keanggotaan yang telah dibuat sebelumnya untuk mendapatkan derajat keanggotaannya. Kemudian proses ini akan dilanjutkan dengan mengambil nilai derajat keanggotaan minimum dari ketiga variabel masukan tersebut.

3.2.3.5 Proses Sistem Inferensi

Setelah tahap fuzzyfikasi, akan dilanjutkan pada proses sistem inferensi dimana akan dilakukan penarikan kesimpulan dari hasil nilai minimum derajat keanggotaan pada tiap-tiap *rule* yang telah dibangun. Kemudian penarikan kesimpulan akan dilakukan terhadap fungsi keanggotaan variabel keluaran (*output*) berupa harga yang telah ditentukan sebelumnya. Daftar aturan-aturan (*rule*) yang dibangun didalam sistem pada perhitungan estimasi harga dan perhitungan harga dapat dilihat pada **Tabel 3.4 dan Tabel 3.5**.

3.2.3.6 Proses Defuzzyfikasi

Setelah dilakukan proses penarikan kesimpulan akan dilakukan proses defuzzyfikasi untuk mengembalikan nilai *fuzzy* menjadi nilai tegas. Pada tahap ini akan digunakan metode *Weighted Average Method* dengan rumus sebagai berikut :

$$Z = \frac{\sum(a_{_p_i} * z_i)}{\sum a_{_p_i}} \quad (3.13)$$

Keterangan :

Z = Hasil *Defuzzyfication* metode rata-rata terbobot

$a_{_p_i}$ = Nilai minimum derajat keanggotaan

z_i = Nilai *crisp* hasil inferensi rule

i = Jumlah aturan *fuzzy*

Tabel 3.4 Aturan (*Rule*) Perhitungan Estimasi Harga

No.	Tingkat Kesulitan	Tanggal Transaksi	Harga
1	Mudah	Tidak Mendadak	Murah
2	Mudah	Sedang	Murah
3	Mudah	Mendadak	Sedang
4	Sedang	Tidak Mendadak	Murah
5	Sedang	Sedang	Sedang
6	Sedang	Mendadak	Sedang
7	Sulit	Tidak Mendadak	Sedang
8	Sulit	Sedang	Mahal
9	Sulit	Mendadak	Mahal

Tabel 3.5 Aturan (*Rule*) Perhitungan Harga

No	Tingkat Kesulitan	Waktu	Jarak	Harga
1	Mudah	Mendadak	Dekat	Murah
2	Mudah	Mendadak	Sedang	Sedang
3	Mudah	Mendadak	Jauh	Sedang
4	Mudah	Sedang	Dekat	Murah
5	Mudah	Sedang	Sedang	Sedang
6	Mudah	Sedang	Jauh	Sedang
7	Mudah	Tidak Mendadak	Dekat	Murah
8	Mudah	Tidak Mendadak	Sedang	Murah
9	Mudah	Tidak Mendadak	Jauh	Sedang
10	Sedang	Mendadak	Dekat	Sedang
11	Sedang	Mendadak	Sedang	Sedang
12	Sedang	Mendadak	Jauh	Mahal
13	Sedang	Sedang	Dekat	Sedang
14	Sedang	Sedang	Sedang	Sedang
15	Sedang	Sedang	Jauh	Mahal
16	Sedang	Tidak Mendadak	Dekat	Murang
17	Sedang	Tidak Mendadak	Sedang	Sedang
18	Sedang	Tidak Mendadak	Jauh	Sedang
19	Sulit	Mendadak	Dekat	Sedang
20	Sulit	Mendadak	Sedang	Mahal
21	Sulit	Mendadak	Jauh	Mahal
22	Sulit	Sedang	Dekat	Sedang
23	Sulit	Sedang	Sedang	Mahal
24	Sulit	Sedang	Jauh	Mahal
25	Sulit	Tidak Mendadak	Dekat	Sedang
26	Sulit	Tidak Mendadak	Sedang	Sedang
27	Sulit	Tidak Mendadak	Jauh	Mahal

BAB IV IMPLEMENTASI

Pada bab ini diuraikan mengenai implementasi perangkat lunak dari rancangan metode yang telah dibahas pada Bab III meliputi kode program dalam perangkat lunak. Selain itu, implementasi dari tiap proses, parameter masukan, keluaran, dan beberapa keterangan yang berhubungan dengan program juga akan dijelaskan pada bab ini.

4.1 Lingkungan Implementasi

Spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam implementasi ini ditampilkan pada **Tabel 4.1**.

Tabel 4.1 Lingkungan Perancangan Perangkat Lunak

Perangkat	Spesifikasi
Perangkat keras	Prosesor: Intel® Core™ i7-3537U CPU @ 2.00GHz(4CPUs), ~2.0GHz Memori: 8.00 GB
Perangkat lunak	Sistem Operasi: Microsoft Windows 8.1 64-bit Pro Perangkat Pengembang: Android Studio Sublime Text 2 Perangkat Pembantu: NetBeans IDE

4.2 Implementasi Metode pada *NetBeans IDE*

Implementasi metode juga dilakukan pada perangkat pembantu *NetBeans IDE* untuk memudahkan pengembang melakukan evaluasi hasil dari perhitungan metode *Fuzzy Tsukamoto*. Metode diimplementasikan dalam bentuk aplikasi sederhana dimana masukan (input) berupa variabel yang ditentukan dan keluaran (*output*) berupa hasil dari langkah-langkah perhitungan beserta hasil akhir perhitungan metode tersebut.

4.2.1 Implementasi Fungsi Keanggotaan

Fungsi keanggotaan akan diimplementasikan untuk mendukung proses fuzzyfikasi. Batas-batas seperti semesta pembicaraan dan domain yang telah ditentukan sebelumnya akan diimplementasikan pada masing-masing ketentuan pada fungsi keanggotaan yang akan dibangun. Terdapat 4 fungsi keanggotaan yang akan dibangun yaitu fungsi keanggotaan tingkat kesulitan, waktu dan jarak sebagai masukan (*input*) dan fungsi keanggotaan harga sebagai keluaran (*output*).

4.2.1.1 Fungsi Keanggotaan Tingkat Kesulitan

Fungsi keanggotaan tingkat kesulitan memiliki semesta pembicaraan [1,13] yang berarti masukan dari variabel tingkat kesulitan berupa nilai 1 hingga 13. Fungsi keanggotaan tingkat kesulitan memiliki 3 domain yaitu MUDAH [1,6], SEDANG [4,9] dan SULIT [9,13]. Pada domain mudah masukan tingkat kesulitan 1 – 4 memiliki derajat keanggotaan sebesar 1, sedangkan untuk masukan tingkat kesulitan 5 memiliki derajat keanggotaan sebesar 0,5 dan masukan tingkat kesulitan 6 memiliki derajat keanggotaan sebesar 0. Implementasi dari fungsi keanggotaan tingkat kesulitan pada domain mudah, sedang dan sulit dapat diamati pada **Kode Sumber 4.1 – Kode Sumber 4.3** sebagai berikut.

```
public static double mudah() {
    if (T_Kesulitan >= 1 && T_Kesulitan <= 4)
        return 1;
    else if (T_Kesulitan > 4 && T_Kesulitan < 6)
        return (6 - T_Kesulitan) / 2;
    else
        return 0;
}
```

Kode Sumber 4.1 Implementasi Fungsi Keanggotaan Tingkat Kesulitan Domain Mudah

Pada domain sedang masukan tingkat kesulitan 6 – 9 memiliki derajat keanggotaan sebesar 1, sedangkan untuk masukan tingkat kesulitan 5 memiliki derajat keanggotaan sebesar 0,5 dan masukan tingkat kesulitan dimana tingkat kesulitan ≤ 4 dan > 9 memiliki derajat keanggotaan sebesar 0.

```
public static double sedang(){
    if(T_Kesulitan <= 4 && T_Kesulitan > 9)
        return 0;
    else if(T_Kesulitan > 4 && T_Kesulitan <= 6)
        return (T_Kesulitan - 4) / 2;
    else if(T_Kesulitan > 6 && T_Kesulitan <= 9)
        return 1;
    else
        return 0;
}
```

Kode Sumber 4.2 Implementasi Fungsi Keanggotaan Tingkat Kesulitan Domain Sedang

Pada domain sulit masukan tingkat kesulitan 12 – 13 memiliki derajat keanggotaan sebesar 1, sedangkan untuk masukan tingkat kesulitan 10 – 11 memiliki derajat keanggotaan hasil dari perhitungan $(T_Kesulitan - 9) / 3$ dan masukan tingkat kesulitan dimana tingkat kesulitan ≤ 9 dan > 13 memiliki derajat keanggotaan sebesar 0.

```
public static double sulit(){
    if(T_Kesulitan <= 9 && T_Kesulitan > 13)
        return 0;
    else if(T_Kesulitan > 9 && T_Kesulitan <= 12)
        return (T_Kesulitan - 9) / 3;
    else if(T_Kesulitan > 12 && T_Kesulitan <= 13)
        return 1;
    else
        return 0;
}
```

Kode Sumber 4.3 Implementasi Fungsi Keanggotaan Tingkat Kesulitan Domain Sulit

4.2.1.2 Fungsi Keanggotaan Waktu

Fungsi keanggotaan waktu memiliki semesta pembicaraan $[0,14]$ yang berarti masukan dari variabel waktu berupa nilai 0 hingga 14. Fungsi keanggotaan waktu memiliki 3 domain yaitu MENDADAK $[0,5]$, SEDANG $[3,7]$ dan TIDAK MENDADAK $[7,14]$. Pada domain mendadak masukan waktu 0 – 3 memiliki derajat keanggotaan sebesar 1, sedangkan untuk masukan waktu 4 memiliki derajat keanggotaan sebesar 0,5 dan masukan waktu 5 memiliki derajat keanggotaan sebesar 0.

```
public static double mendadak(){
    if(waktu >= 0 && waktu <= 3)
        return 1;
    else if (waktu > 3 && waktu < 5)
        return (5 - waktu) / 2;
    else
        return 0;
}
```

Kode Sumber 4.4 Implementasi Fungsi Keanggotaan Waktu Domain Mendadak

Pada domain sedang masukan waktu 5 – 7 memiliki derajat keanggotaan sebesar 1, sedangkan untuk masukan waktu 4 memiliki derajat keanggotaan sebesar 0,5 dan masukan waktu dimana waktu ≤ 3 dan > 7 memiliki derajat keanggotaan sebesar 0.

```
public static double sedang(){
    if(waktu <= 3 && waktu > 7)
        return 0;
    else if(waktu > 3 && waktu <= 5)
        return (waktu - 3) / 2;
    else if(waktu > 5 && waktu <= 7)
        return 1;
    else
        return 0;
}
```

Kode Sumber 4.5 Implementasi Fungsi Keanggotaan Waktu Domain Sedang

Pada domain tidak mendadak masukan waktu 10 – 14 memiliki derajat keanggotaan sebesar 1, sedangkan untuk masukan waktu berupa 8 – 9 memiliki derajat keanggotaan hasil dari perhitungan $(\text{waktu} - 6) / 2$ dan masukan waktu dimana nilai waktu ≤ 7 dan > 14 memiliki derajat keanggotaan sebesar 0.

```
public static double tdkmendadak(){
    if(waktu <= 7)
        return 0;
    else if(waktu > 7 && waktu < 10)
        return (waktu - 6) / 2;
    else if(waktu >= 10 && waktu <= 14)
        return 1;
    else
        return 0;
}
```

Kode Sumber 4.6 Implementasi Fungsi Keanggotaan Waktu Domain Tidak Mendadak

4.2.1.3 Fungsi Keanggotaan Jarak

Fungsi keanggotaan jarak memiliki semesta pembicaraan [0,35000] yang berarti masukan dari variabel jarak berupa nilai 0 hingga 35000. Fungsi keanggotaan jarak memiliki 3 domain yaitu DEKAT [0,5000], SEDANG [3000,7500], JAUH [7500,35000]. Pada domain dekat masukan jarak 0 – 3000 memiliki derajat keanggotaan sebesar 1, sedangkan untuk masukan jarak dari rentan 3001 - 4999 memiliki derajat keanggotaan hasil dari perhitungan $(5000 - \text{jarak}) / 2000$ dan masukan jarak 5000 memiliki derajat keanggotaan sebesar 0. Implementasi dari fungsi keanggotaan jarak pada domain dekat, sedang dan jauh dapat diamati pada **Kode Sumber 4.7 – Kode Sumber 4.9** yang ada seperti pada berikut ini.

```

public static double dekat(){
    if(jarak >= 0 && jarak <= 3000)
        return 1;
    else if (jarak > 3000 && jarak < 5000)
        return (5000 - jarak) / 2000;
    else
        return 0;
}

```

**Kode Sumber 4.7 Implementasi Fungsi Keanggotaan Jarak
Domain Dekat**

Pada domain sedang masukan jarak 5000 - 7500 memiliki derajat keanggotaan sebesar 1, sedangkan untuk masukan jarak 3001 - 4999 memiliki derajat keanggotaan berupa hasil dari perhitungan $(\text{jarak} - 3000) / 2000$ dan masukan waktu dimana waktu ≤ 3000 dan > 7500 memiliki derajat keanggotaan 0.

```

public static double sedang(){
    if(jarak <= 3000 && jarak > 7500)
        return 0;
    else if(jarak > 3000 && jarak <= 5000)
        return (jarak - 3000) / 2000;
    else if(jarak > 5000 && jarak <= 7500)
        return 1;
    else
        return 0;
}

```

**Kode Sumber 4.8 Implementasi Fungsi Keanggotaan Jarak
Domain Sedang**

Pada domain jauh masukan jarak 9000 - 35000 memiliki derajat keanggotaan sebesar 1, sedangkan untuk masukan jarak berupa 7501 - 9000 memiliki derajat keanggotaan hasil dari perhitungan $(\text{jarak} - 7500) / 1500$ dan masukan jarak dimana nilai jarak ≤ 7500 dan > 35000 memiliki derajat keanggotaan sebesar 0.

```

public static double jauh(){
    if(jarak <= 7500 && jarak > 35000)
        return 0;
    else if(jarak > 7500 && jarak <= 9000)
        return (jarak - 7500) / 1500;
    else if(jarak > 9000 && jarak <= 35000)
        return 1;
    else
        return 0;
}

```

Kode Sumber 4.9 Implementasi Fungsi Keanggotaan Jarak Domain Jauh

4.2.1.4 Fungsi Keanggotaan Harga

Fungsi keanggotaan harga memiliki semesta pembicaraan [35000,100000] yang berarti keluaran dari variabel harga berupa nilai 0 hingga 100000. Fungsi keanggotaan harga memiliki 3 domain yaitu MURAH [35000,60000], SEDANG [45000,80000] dan MAHAL [80000,100000]. Pada domain murah keluaran 35000 - 45000 memiliki derajat keanggotaan sebesar 1, sedangkan untuk keluaran harga 45001 - 59999 memiliki derajat keanggotaan hasil dari perhitungan $(60000 - \text{harga}) / 15000$ dan keluaran harga 60000 memiliki derajat keanggotaan sebesar 0. Implementasi dari fungsi keanggotaan harga dapat diamati pada **Kode Sumber 4.10 – Kode Sumber 4.15** sebagai berikut.

```

public static double murah(){
    if(harga >= 35000 && harga <= 45000)
        return 1;
    else if (harga > 45000 && harga < 60000)
        return (60000 - harga) / 15000;
    else
        return 0;
}

```

Kode Sumber 4.10 Implementasi Fungsi Keanggotaan Harga Domain Murah (1)

```
public static double murah(double alfa){
    return(60000 - (alfa*15000));
}
```

Kode Sumber 4.11 Implementasi Fungsi Keanggotaan Harga Domain Murah (2)

Pada domain sedang keluaran 60000 – 80000 memiliki derajat keanggotaan sebesar 1, sedangkan untuk keluaran harga 45001 - 59999 memiliki derajat keanggotaan berupa hasil dari perhitungan $(\text{harga} - 45000) / 15000$ dan keluaran harga dimana harga ≤ 45000 dan > 80000 memiliki derajat keanggotaan 0.

```
public static double sedang(){
    if(harga >= 60000 && harga <= 80000)
        return 1;
    else if(harga > 45000 && harga < 60000)
        return (harga - 45000) / 15000;
    else if(harga <= 45000 && harga > 80000)
        return 0;
    else
        return 0;
}
```

Kode Sumber 4.12 Implementasi Fungsi Keanggotaan Harga Domain Sengah (1)

```
public static double sedang(double alfa){
    return(45000 + (alfa*15000));
}
```

Kode Sumber 4.13 Implementasi Fungsi Keanggotaan Harga Domain Sengah (2)

Pada domain sedang keluaran 90000 – 100000 memiliki derajat keanggotaan sebesar 1, sedangkan untuk keluaran harga 80001 - 89999 memiliki derajat keanggotaan berupa hasil dari perhitungan $(\text{harga} - 80000) / 10000$ dan keluaran harga dimana harga ≤ 80000 dan > 100000 memiliki derajat keanggotaan 0.

```
public static double mahal(){
    if(harga <= 80000 && harga > 100000)
        return 0;
    else if(harga > 80000 && harga < 90000)
        return (harga - 80000) / 10000;
    else if(harga >= 90000 && harga <= 100000)
        return 1;
    else
        return 0;
}
```

Kode Sumber 4.14 Implementasi Fungsi Keanggotaan Harga Domain Mahal (1)

```
public static double mahal(double alfa){
    return(80000 + (alfa*10000));
}
```

Kode Sumber 4.15 Implementasi Fungsi Keanggotaan Harga Domain Mahal (2)

4.2.2 Implementasi *Fuzzyfication* dan *Inference System*

Proses fuzzyfikasi dilakukan pada tiap-tiap aturan (*rule*) yang telah dibangun pada sistem. Terdapat 27 aturan (*rule*) yang dibangun ke dalam sistem. Jumlah aturan tersebut didapatkan dari hasil perhitungan $\text{Jumlah_Domain}^{\text{Jumlah_Variabel}}$. Proses ini akan dilakukan terhadap masukan yang diberikan *user* berupa tingkat kesulitan, waktu dan jarak pada fungsi `hitungX()`.

Pada fungsi ini proses dimulai dengan mengambil masukan dan memanggil kelas masukan yang berkaitan untuk mendapatkan nilai *fuzzy* atau derajat keanggotaannya. Kemudian akan diambil nilai terendah dari ketiga masukan tersebut menggunakan metode

MIN dengan mengambil nilai terendah antara waktu dan jarak. Hasil tersebut kemudian dibandingkan kembali dengan nilai *fuzzy* tingkat kesulitan untuk mendapatkan nilai minimum dari ketiga masukan tersebut. Setelah didapatkan nilai minimum dari ketiga variabel pada masing-masing *rule*, akan dilakukan penarikan kesimpulan pada fungsi `hitungY()`.

```
public static void hitungX(){
    xHarga[0] = Math.min(TingkatKesulitan.mudah(),
        Math.min(Waktu.mendadak(), Jarak.dekat()));
    xHarga[1] = Math.min(TingkatKesulitan.mudah(),
        Math.min(Waktu.mendadak(), Jarak.sedang()));
    xHarga[2] = Math.min(TingkatKesulitan.mudah(),
        Math.min(Waktu.mendadak(), Jarak.jauh()));
    xHarga[3] = Math.min(TingkatKesulitan.mudah(),
        Math.min(Waktu.sedang(), Jarak.dekat()));
    xHarga[4] = Math.min(TingkatKesulitan.mudah(),
        Math.min(Waktu.sedang(), Jarak.sedang()));
    xHarga[5] = Math.min(TingkatKesulitan.mudah(),
        Math.min(Waktu.sedang(), Jarak.jauh()));
    xHarga[6] = Math.min(TingkatKesulitan.mudah(),
        Math.min(Waktu.tdkmendadak(), Jarak.dekat()));
    xHarga[7] = Math.min(TingkatKesulitan.mudah(),
        Math.min(Waktu.tdkmendadak(), Jarak.sedang()));
    xHarga[8] = Math.min(TingkatKesulitan.mudah(),
        Math.min(Waktu.tdkmendadak(), Jarak.jauh()));
    xHarga[9] = Math.min(TingkatKesulitan.sedang(),
        Math.min(Waktu.mendadak(), Jarak.dekat()));
    xHarga[10] = Math.min(TingkatKesulitan.sedang(),
        Math.min(Waktu.mendadak(), Jarak.sedang()));
    xHarga[11] = Math.min(TingkatKesulitan.sedang(),
        Math.min(Waktu.mendadak(), Jarak.jauh()));
    xHarga[12] = Math.min(TingkatKesulitan.sedang(),
        Math.min(Waktu.sedang(), Jarak.dekat()));
    xHarga[13] = Math.min(TingkatKesulitan.sedang(),
        Math.min(Waktu.sedang(), Jarak.sedang()));
    xHarga[14] = Math.min(TingkatKesulitan.sedang(),
        Math.min(Waktu.sedang(), Jarak.jauh()));
    xHarga[15] = Math.min(TingkatKesulitan.sedang(),
        Math.min(Waktu.tdkmendadak(), Jarak.dekat()));
```

Kode Sumber 4.16 Implementasi Fungsi `hitungX()` (1)

```

xHarga[16] = Math.min(TingkatKesulitan.sedang(),
    Math.min(Waktu.tdkmendadak(), Jarak.sedang()));
xHarga[17] = Math.min(TingkatKesulitan.sedang(),
    Math.min(Waktu.tdkmendadak(), Jarak.jauh()));
xHarga[18] = Math.min(TingkatKesulitan.sulit(),
    Math.min(Waktu.mendadak(), Jarak.dekat()));
xHarga[19] = Math.min(TingkatKesulitan.sulit(),
    Math.min(Waktu.mendadak(), Jarak.sedang()));
xHarga[20] = Math.min(TingkatKesulitan.sulit(),
    Math.min(Waktu.mendadak(), Jarak.jauh()));
xHarga[21] = Math.min(TingkatKesulitan.sulit(),
    Math.min(Waktu.sedang(), Jarak.dekat()));
xHarga[22] = Math.min(TingkatKesulitan.sulit(),
    Math.min(Waktu.sedang(), Jarak.sedang()));
xHarga[23] = Math.min(TingkatKesulitan.sulit(),
    Math.min(Waktu.sedang(), Jarak.jauh()));
xHarga[24] = Math.min(TingkatKesulitan.sulit(),
    Math.min(Waktu.tdkmendadak(), Jarak.dekat()));
xHarga[25] = Math.min(TingkatKesulitan.sulit(),
    Math.min(Waktu.tdkmendadak(), Jarak.sedang()));
xHarga[26] = Math.min(TingkatKesulitan.sulit(),
    Math.min(Waktu.tdkmendadak(), Jarak.jauh()));
}

```

Kode Sumber 4.17 Implementasi Fungsi `hitungX()` (2)

Pada fungsi `hitungY()` nilai minimum hasil dari perhitungan pada fungsi `hitungX()` akan diolah kembali pada fungsi keanggotaan harga dengan cara memanggil domain kelas harga yang sesuai dengan penarikan kesimpulan pada masing-masing *rule*.

```

public static void hitungY(){
    yHarga[0] = Harga.murah(xHarga[0]);
    yHarga[1] = Harga.sedang(xHarga[1]);
    yHarga[2] = Harga.sedang(xHarga[2]);
    yHarga[3] = Harga.murah(xHarga[3]);
    yHarga[4] = Harga.sedang(xHarga[4]);
    yHarga[5] = Harga.sedang(xHarga[5]);
    yHarga[6] = Harga.murah(xHarga[6]);
    yHarga[7] = Harga.murah(xHarga[7]);
    yHarga[8] = Harga.sedang(xHarga[8]);
}

```

Kode Sumber 4.18 Implementasi Fungsi `hitungY()` (1)

```

yHarga[9] = Harga.sedang(xHarga[9]);
yHarga[10] = Harga.sedang(xHarga[10]);
yHarga[11] = Harga.mahal(xHarga[11]);
yHarga[12] = Harga.sedang(xHarga[12]);
yHarga[13] = Harga.sedang(xHarga[13]);
yHarga[14] = Harga.mahal(xHarga[14]);
yHarga[15] = Harga.murah(xHarga[15]);
yHarga[16] = Harga.sedang(xHarga[16]);
yHarga[17] = Harga.sedang(xHarga[17]);
yHarga[18] = Harga.sedang(xHarga[18]);
yHarga[19] = Harga.mahal(xHarga[19]);
yHarga[20] = Harga.mahal(xHarga[20]);
yHarga[21] = Harga.sedang(xHarga[21]);
yHarga[22] = Harga.mahal(xHarga[22]);
yHarga[23] = Harga.mahal(xHarga[23]);
yHarga[24] = Harga.sedang(xHarga[24]);
yHarga[25] = Harga.sedang(xHarga[25]);
yHarga[26] = Harga.mahal(xHarga[26]);
}

```

Kode Sumber 4.19 Implementasi Fungsi *hitungY()* (2)

4.2.3 Implementasi *Defuzzyfication*

Proses *defuzzyfication* akan dilakukan pada fungsi *defuzzyfikasi()*. Pada fungsi ini akan dilakukan penjumlahan hasil dari perhitungan fungsi *hitungX()* berupa nilai minimum dari derajat keanggotaan yang dikalikan dengan nilai *crisp* hasil perhitungan fungsi *hitungY()*. Kemudian hasil penjumlahan dari perhitungan tersebut akan dibagi dengan jumlah nilai dari nilai minimum derajat keanggotaan pada fungsi *hitungX()*.

```

public static double defuzzyfikasi(){
    double atas = 0, bawah = 0;
    for(int i = 0 ; i < 26 ; i++){
        atas += (xHarga[i] * yHarga[i]);
        bawah += xHarga[i];
    }
    System.out.println(atas);
    System.out.println(atas/bawah);
    return(atas/bawah);
}

```

Kode Sumber 4.20 Implementasi Fungsi *defuzzyfikasi()*

4.3 Implementasi Metode pada Android Studio

Hasil dari implementasi metode *Fuzzy Tsukamoto* pada *NetBeans IDE* diimplementasikan dan dikembangkan kembali pada *Android Studio* untuk mendukung proses bisnis yang ada pada aplikasi “*Finding Tutor*”. Terdapat 2 implementasi pada aplikasi yaitu perhitungan untuk estimasi harga dan perhitungan untuk harga final dari penyedia jasa (tutor). Berbeda dengan perhitungan harga yang dilakukan dengan menggunakan 3 variabel masukan, pada perhitungan estimasi harga hanya menggunakan 2 variabel masukan berupa tingkat kesulitan dan waktu.

4.3.1 Implementasi Persiapan Masukan Sistem *Fuzzy*

Terdapat 3 masukan user yang akan digunakan pada sistem yaitu tingkat kesulitan, waktu dan jarak. Masukan tingkat kesulitan yang dimasukkan *user* berupa tingkat pendidikan pada umumnya, sehingga masukan tersebut harus diubah menjadi nilai untuk tiap-tiap tingkatan pendidikan yang dapat digunakan pada sistem.

```
TingkatKesulitan tingkatKesulitan = new
TingkatKesulitan();
if(dataMurid.getKelas_pencarian().
    matches("SD - Kelas 1")){
    kesulitan = 1;
}
else if(dataMurid.getKelas_pencarian().
    matches("SD - Kelas 2")){
    kesulitan = 2;
}
else if(dataMurid.getKelas_pencarian().
    matches("SD - Kelas 3")){
    kesulitan = 3;
}
else if(dataMurid.getKelas_pencarian().
    matches("SD - Kelas 4")){
    kesulitan = 4;
}
```

Kode Sumber 4.21 Implementasi Masukan Tingkat Kesulitan (1)

```

else if (dataMurid.getKelas_pencarian() .
    matches("SD - Kelas 5")){
    kesulitan = 5;
}
else if (dataMurid.getKelas_pencarian() .
    matches("SD - Kelas 6")){
    kesulitan = 6;
}
else if (dataMurid.getKelas_pencarian() .
    matches("SMP - Kelas 7")){
    kesulitan = 7;
}
else if (dataMurid.getKelas_pencarian() .
    matches("SMP - Kelas 8")){
    kesulitan = 8;
}
else if (dataMurid.getKelas_pencarian() .
    matches("SMP - Kelas 9")){
    kesulitan = 9;
}
else if (dataMurid.getKelas_pencarian() .
    matches("SMA - Kelas 10")){
    kesulitan = 10;
}
else if (dataMurid.getKelas_pencarian() .
    matches("SMA - Kelas 11")){
    kesulitan = 11;
}
else if (dataMurid.getKelas_pencarian() .
    matches("SMA - Kelas 12")){
    kesulitan = 12;
}
else if (dataMurid.getKelas_pencarian() .
    matches("UMUM")){
    kesulitan = 13;
}

```

Kode Sumber 4.22 Implementasi Masukan Tingkat Kesulitan (2)

Masukan *user* waktu berupa tanggal dimana transaksi mengajar akan dilakukan, sehingga harus dilakukan perhitungan selisih tanggal transaksi tersebut dengan tanggal yang ada pada sistem.

Langkah yang pertama adalah menghilangkan “/” yang ada pada format tanggal masukan dan memasukkannya pada masing-masing index variabel untuk membedakan tanggal, bulan dan tahun. Kemudian sistem akan mengambil tanggal yang ada pada sistem. Setelah kedua tanggal telah didapatkan, tanggal masukan akan di set sesuai dengan format pada tanggal sistem. Setelah itu keduanya akan dikurangkan untuk mendapatkan selisih dari kedua tanggal tersebut dan dibagi untuk mendapatkan nilai selisih tersebut.

```
int tanggal = Integer.parseInt(dataMurid.
    getTanggal_pencarian().split("/") [0]);
int bulan    = Integer.parseInt(dataMurid.
    getTanggal_pencarian().split("/") [1]);
int tahun    = Integer.parseInt(dataMurid.
    getTanggal_pencarian().split("/") [2]);

Calendar today = Calendar.getInstance();
Calendar tanggalLes = Calendar.getInstance();
tanggalLes.set(Calendar.DAY_OF_MONTH, tanggal);
tanggalLes.set(Calendar.MONTH, bulan-1);
tanggalLes.set(Calendar.YEAR, tahun);

long diff = tanggalLes.getTimeInMillis()
    -today.getTimeInMillis();
long diffDay = diff/(24*60*60*1000);

Log.d("tanggal", String.valueOf(diffDay));

Waktu waktu = new Waktu();
waktu.setWaktu(diffDay);
```

Kode Sumber 4.23 Implementasi Masukan Waktu

Masukan jarak berupa jarak diantara alamat tutor dan murid. Langkah pertama adalah mendapatkan *latitude* dan *longitude* kedua alamat tersebut dengan *Geocoder*. Kemudian sistem akan melakukan *request* terhadap *Google API* dengan *key* yang telah didaftarkan sebelumnya. Response yang akan diakses yaitu *routes*, *legs*, *distance* yang akan menghasilkan nilai berupa jarak diantara kedua alamat tersebut berdasarkan dengan rute jalan yang ada dan

kemudian menyimpannya sebagai *object* untuk mempermudah penggunaan data pada fungsi lainnya.

```

Mendapatkan Latitude dan Longitude
-----
geocoder = new Geocoder(getBaseContext());
try {
    List<Address> listMurid =
        geocoder.getFromLocationName(alamat, 1);
    Address alamatMurid = listMurid.get(0);
    latMurid = alamatMurid.getLatitude();
    longMurid = alamatMurid.getLongitude();

    List<Address> listTutor =
        geocoder.getFromLocationName(alamatTutordb, 1);
    Address alamatTutor = listTutor.get(0);
    latTutor = alamatTutor.getLatitude();
    longTutor = alamatTutor.getLongitude();
} catch (IOException e) {
    e.printStackTrace(); }

Request Google API
-----
StringRequest stringRequest = new
StringRequest(Request.Method.GET,
    "https://maps.googleapis.com/maps/api/directions/json?" +
    "origin=" +latTutor+ "," +longTutor+
    "&destination="+latMurid+"," +longMurid+
    "&key=AIzaSyCwH6FT975GOvqRVaf_-rmp429uGgFXhR0", new
Response.Listener<String>() {
    public void onResponse(String response) {
        try {
            JSONObject jsonObject = new JSONObject(response);
            if (jsonObject.length()>0){
                JSONArray arrayDistanceMap =
                    jsonObject.getJSONArray("routes");
                JSONObject objectDistanceMap =
                    arrayDistanceMap.getJSONObject(0);
                JSONArray jarak =
                    objectDistanceMap.getJSONArray("legs");
                JSONObject objectDistance =
                    jarak.getJSONObject(0);
                JSONObject jarakFinal =
                    objectDistance.getJSONObject("distance");
                getJarak =
                    Float.valueOf(jarakFinal.getString("value"));

```

Kode Sumber 4.24 Implementasi Masukan Jarak

4.3.2 Implementasi Fungsi Keanggotaan

Fungsi keanggotaan akan diimplementasikan untuk mendukung proses fuzzyfikasi. Batas-batas seperti semesta pembicaraan dan domain yang telah ditentukan sebelumnya akan diimplementasikan pada masing-masing ketentuan pada fungsi keanggotaan yang akan dibangun. Terdapat 4 fungsi keanggotaan yang akan dibangun yaitu fungsi keanggotaan tingkat kesulitan, waktu dan jarak sebagai masukan (*input*) dan fungsi keanggotaan harga sebagai keluaran (*output*).

4.3.2.1 Fungsi Keanggotaan Tingkat Kesulitan

Fungsi keanggotaan tingkat kesulitan memiliki semesta pembicaraan [1,13] yang berarti masukan dari variabel tingkat kesulitan berupa nilai 1 hingga 13. Fungsi keanggotaan tingkat kesulitan memiliki 3 domain yaitu MUDAH [1,6], SEDANG [4,9] dan SULIT [9,13]. Pada domain mudah masukan tingkat kesulitan 1 – 4 memiliki derajat keanggotaan sebesar 1, sedangkan untuk masukan tingkat kesulitan 5 memiliki derajat keanggotaan sebesar 0,5 dan masukan tingkat kesulitan 6 memiliki derajat keanggotaan sebesar 0. Implementasi dari fungsi keanggotaan tingkat kesulitan pada domain mudah, sedang dan sulit dapat diamati pada **Kode Sumber 4.25 – Kode Sumber 4.27** sebagai berikut.

```
public static double mudah() {
    if(T_Kesulitan >= 1 && T_Kesulitan <= 4)
        return 1;
    else if (T_Kesulitan > 4 && T_Kesulitan < 6)
        return (6 - T_Kesulitan) / 2;
    else
        return 0;
}
```

Kode Sumber 4.25 Implementasi Fungsi Keanggotaan Tingkat Kesulitan Domain Mudah

Pada domain sedang masukan tingkat kesulitan 6 – 9 memiliki derajat keanggotaan sebesar 1, sedangkan untuk masukan tingkat kesulitan 5 memiliki derajat keanggotaan sebesar 0,5 dan masukan tingkat kesulitan dimana tingkat kesulitan ≤ 4 dan > 9 memiliki derajat keanggotaan sebesar 0.

```
public static double sedang(){
    if(T_Kesulitan <= 4 && T_Kesulitan > 9)
        return 0;
    else if(T_Kesulitan > 4 && T_Kesulitan <= 6)
        return (T_Kesulitan - 4) / 2;
    else if(T_Kesulitan > 6 && T_Kesulitan <= 9)
        return 1;
    else
        return 0;
}
```

Kode Sumber 4.26 Implementasi Fungsi Keanggotaan Tingkat Kesulitan Domain Sedang

Pada domain sulit masukan tingkat kesulitan 12 – 13 memiliki derajat keanggotaan sebesar 1, sedangkan untuk masukan tingkat kesulitan 10 – 11 memiliki derajat keanggotaan hasil dari perhitungan $(T_Kesulitan - 9) / 3$ dan masukan tingkat kesulitan dimana tingkat kesulitan ≤ 9 dan > 13 memiliki derajat keanggotaan sebesar 0.

```
public static double sulit(){
    if(T_Kesulitan <= 9 && T_Kesulitan > 13)
        return 0;
    else if(T_Kesulitan > 9 && T_Kesulitan <= 12)
        return (T_Kesulitan - 9) / 3;
    else if(T_Kesulitan > 12 && T_Kesulitan <= 13)
        return 1;
    else
        return 0;
}
```

Kode Sumber 4.27 Implementasi Fungsi Keanggotaan Tingkat Kesulitan Domain Sulit

4.3.2.2 Fungsi Keanggotaan Waktu

Fungsi keanggotaan waktu memiliki semesta pembicaraan $[0,14]$ yang berarti masukan dari variabel waktu berupa nilai 0 hingga 14. Fungsi keanggotaan waktu memiliki 3 domain yaitu MENDADAK $[0,5]$, SEDANG $[3,7]$ dan TIDAK MENDADAK $[7,14]$. Pada domain mendadak masukan waktu 0 – 3 memiliki derajat keanggotaan sebesar 1, sedangkan untuk masukan waktu 4 memiliki derajat keanggotaan sebesar 0,5 dan masukan waktu 5 memiliki derajat keanggotaan sebesar 0.

```
public static double mendadak(){
    if(waktu >= 0 && waktu <= 3)
        return 1;
    else if (waktu > 3 && waktu < 5)
        return (5 - waktu) / 2;
    else
        return 0;
}
```

**Kode Sumber 4.28 Implementasi Fungsi Keanggotaan Waktu
Domain Mendadak**

Pada domain sedang masukan waktu 5 – 7 memiliki derajat keanggotaan sebesar 1, sedangkan untuk masukan waktu 4 memiliki derajat keanggotaan sebesar 0,5 dan masukan waktu dimana waktu ≤ 3 dan > 7 memiliki derajat keanggotaan sebesar 0.

```
public static double sedang(){
    if(waktu <= 3 && waktu > 7)
        return 0;
    else if(waktu > 3 && waktu <= 5)
        return (waktu - 3) / 2;
    else if(waktu > 5 && waktu <= 7)
        return 1;
    else
        return 0;
}
```

**Kode Sumber 4.29 Implementasi Fungsi Keanggotaan Waktu
Domain Sedang**

Pada domain tidak mendadak masukan waktu 10 – 14 memiliki derajat keanggotaan sebesar 1, sedangkan untuk masukan waktu berupa 8 – 9 memiliki derajat keanggotaan hasil dari perhitungan $(\text{waktu} - 6) / 2$ dan masukan waktu dimana nilai waktu ≤ 7 dan > 14 memiliki derajat keanggotaan sebesar 0.

```
public static double tdkmendadak() {
    if(waktu <= 7)
        return 0;
    else if(waktu > 7 && waktu < 10)
        return (waktu - 6) / 2;
    else if(waktu >= 10 && waktu <= 14)
        return 1;
    else
        return 0;
}
```

Kode Sumber 4.30 Implementasi Fungsi Keanggotaan Waktu Domain Tidak Mendadak

4.3.2.3 Fungsi Keanggotaan Jarak

Fungsi keanggotaan jarak memiliki semesta pembicaraan $[0,35000]$ yang berarti masukan dari variabel jarak berupa nilai 0 hingga 35000. Fungsi keanggotaan jarak memiliki 3 domain yaitu DEKAT $[0,5000]$, SEDANG $[3000,7500]$, JAUH $[7500,35000]$. Pada domain dekat masukan jarak 0 – 3000 memiliki derajat keanggotaan sebesar 1, sedangkan untuk masukan jarak dari rentan 3001 - 4999 memiliki derajat keanggotaan hasil dari perhitungan $(5000 - \text{jarak}) / 2000$ dan masukan jarak 5000 memiliki derajat keanggotaan sebesar 0. Implementasi dari fungsi keanggotaan jarak pada domain dekat, sedang dan jauh dapat diamati pada **Kode Sumber 4.31 – Kode Sumber 4.33** yang ada seperti pada berikut ini.

```

public static double dekat(){
    if(jarak >= 0 && jarak <= 3000)
        return 1;
    else if (jarak > 3000 && jarak < 5000)
        return (5000 - jarak) / 2000;
    else
        return 0;
}

```

**Kode Sumber 4.31 Implementasi Fungsi Keanggotaan Jarak
Domain Dekat**

Pada domain sedang masukan jarak 5000 - 7500 memiliki derajat keanggotaan sebesar 1, sedangkan untuk masukan jarak 3001 - 4999 memiliki derajat keanggotaan berupa hasil dari perhitungan $(\text{jarak} - 3000) / 2000$ dan masukan waktu dimana waktu ≤ 3000 dan > 7500 memiliki derajat keanggotaan 0.

```

public static double sedang(){
    if(jarak <= 3000 && jarak > 7500)
        return 0;
    else if(jarak > 3000 && jarak <= 5000)
        return (jarak - 3000) / 2000;
    else if(jarak > 5000 && jarak <= 7500)
        return 1;
    else
        return 0;
}

```

**Kode Sumber 4.32 Implementasi Fungsi Keanggotaan Jarak
Domain Sedang**

Pada domain jauh masukan jarak 9000 - 35000 memiliki derajat keanggotaan sebesar 1, sedangkan untuk masukan jarak berupa 7501 - 9000 memiliki derajat keanggotaan hasil dari perhitungan $(\text{jarak} - 7500) / 1500$ dan masukan jarak dimana nilai jarak ≤ 7500 dan > 35000 memiliki derajat keanggotaan sebesar 0.

```

public static double jauh(){
    if(jarak <= 7500 && jarak > 35000)
        return 0;
    else if(jarak > 7500 && jarak <= 9000)
        return (jarak - 7500) / 1500;
    else if(jarak > 9000 && jarak <= 35000)
        return 1;
    else
        return 0;
}

```

Kode Sumber 4.33 Implementasi Fungsi Keanggotaan Jarak Domain Jauh

4.3.2.4 Fungsi Keanggotaan Harga

Fungsi keanggotaan harga memiliki semesta pembicaraan [35000,100000] yang berarti keluaran dari variabel harga berupa nilai 0 hingga 100000. Fungsi keanggotaan harga memiliki 3 domain yaitu MURAH [35000,60000], SEDANG [45000,80000] dan MAHAL [80000,100000]. Pada domain murah keluaran 35000 - 45000 memiliki derajat keanggotaan sebesar 1, sedangkan untuk keluaran harga 45001 - 59999 memiliki derajat keanggotaan hasil dari perhitungan $(60000 - \text{harga}) / 15000$ dan keluaran harga 60000 memiliki derajat keanggotaan sebesar 0. Implementasi dari fungsi keanggotaan harga dapat diamati pada **Kode Sumber 4.34 – Kode Sumber 4.39** sebagai berikut.

```

public static double murah(){
    if(harga >= 35000 && harga <= 45000)
        return 1;
    else if (harga > 45000 && harga < 60000)
        return (60000 - harga) / 15000;
    else
        return 0;
}

```

Kode Sumber 4.34 Implementasi Fungsi Keanggotaan Harga Domain Murah (1)

```
public static double murah(double alfa){
    return(60000 - (alfa*15000));
}
```

**Kode Sumber 4.35 Implementasi Fungsi Keanggotaan Harga
Domain Murah (2)**

Pada domain sedang keluaran 60000 – 80000 memiliki derajat keanggotaan sebesar 1, sedangkan untuk keluaran harga 45001 - 59999 memiliki derajat keanggotaan berupa hasil dari perhitungan $(\text{harga} - 45000) / 15000$ dan keluaran harga dimana harga ≤ 45000 dan > 80000 memiliki derajat keanggotaan 0.

```
public static double sedang(){
    if(harga >= 60000 && harga <= 80000)
        return 1;
    else if(harga > 45000 && harga < 60000)
        return (harga - 45000) / 15000;
    else if(harga <= 45000 && harga > 80000)
        return 0;
    else
        return 0;
}
```

**Kode Sumber 4.36 Implementasi Fungsi Keanggotaan Harga
Domain Seding (1)**

```
public static double sedang(double alfa){
    return(45000 + (alfa*15000));
}
```

**Kode Sumber 4.37 Implementasi Fungsi Keanggotaan Harga
Domain Seding (2)**

Pada domain sedang keluaran 90000 – 100000 memiliki derajat keanggotaan sebesar 1, sedangkan untuk keluaran harga 80001 - 89999 memiliki derajat keanggotaan berupa hasil dari perhitungan $(\text{harga} - 80000) / 10000$ dan keluaran harga dimana harga ≤ 80000 dan > 100000 memiliki derajat keanggotaan 0.

```
public static double mahal(){
    if(harga <= 80000 && harga > 100000)
        return 0;
    else if(harga > 80000 && harga < 90000)
        return (harga - 80000) / 10000;
    else if(harga >= 90000 && harga <= 100000)
        return 1;
    else
        return 0;
}
```

Kode Sumber 4.38 Implementasi Fungsi Keanggotaan Harga Domain Mahal (1)

```
public static double mahal(double alfa){
    return(80000 + (alfa*10000));
}
```

Kode Sumber 4.39 Implementasi Fungsi Keanggotaan Harga Domain Mahal (2)

4.3.3 Implementasi *Fuzzyfication* dan *Inference System*

Proses fuzzyfikasi dilakukan pada tiap-tiap aturan (*rule*) yang telah dibangun pada sistem. Pada proses fuzzyfikasi dalam perhitungan harga terdapat 27 aturan (*rule*) yang dibangun ke dalam sistem. Jumlah aturan tersebut didapatkan dari hasil perhitungan $\text{Jumlah_Domain}^{\text{Jumlah_Variabel}}$. Proses ini akan dilakukan terhadap masukan yang diberikan *user* berupa tingkat kesulitan, waktu dan jarak pada fungsi `hitungX()`.

Pada fungsi ini proses dimulai dengan mengambil masukan dan memanggil kelas masukan yang berkaitan untuk mendapatkan nilai *fuzzy* atau derajat keanggotaannya. Kemudian akan diambil

nilai terendah dari ketiga masukan tersebut menggunakan metode MIN dengan mengambil nilai terendah antara waktu dan jarak. Hasil tersebut kemudian dibandingkan kembali dengan nilai *fuzzy* tingkat kesulitan untuk mendapatkan nilai minimum dari ketiga masukan tersebut. Setelah didapatkan nilai minimum dari ketiga variabel pada masing-masing *rule*, akan dilakukan penarikan kesimpulan pada fungsi `hitungY()`.

```
public static void hitungX(TingkatKesulitan
tingkatKesulitan1, Waktu waktu1, Jarak jarak1){
    xHarga[0] = Math.min(tingkatKesulitan1.mudah(),
        Math.min(waktu1.mendadak(), jarak1.dekat()));
    xHarga[1] = Math.min(tingkatKesulitan1.mudah(),
        Math.min(waktu1.mendadak(), jarak1.sedang()));
    xHarga[2] = Math.min(tingkatKesulitan1.mudah(),
        Math.min(waktu1.mendadak(), jarak1.jauh()));
    xHarga[3] = Math.min(tingkatKesulitan1.mudah(),
        Math.min(waktu1.sedang(), jarak1.dekat()));
    xHarga[4] = Math.min(tingkatKesulitan1.mudah(),
        Math.min(waktu1.sedang(), jarak1.sedang()));
    xHarga[5] = Math.min(tingkatKesulitan1.mudah(),
        Math.min(waktu1.sedang(), jarak1.jauh()));
    xHarga[6] = Math.min(tingkatKesulitan1.mudah(),
        Math.min(waktu1.tdkmendadak(), jarak1.dekat()));
    xHarga[7] = Math.min(tingkatKesulitan1.mudah(),
        Math.min(waktu1.tdkmendadak(), jarak1.sedang()));
    xHarga[8] = Math.min(tingkatKesulitan1.mudah(),
        Math.min(waktu1.tdkmendadak(), jarak1.jauh()));
    xHarga[9] = Math.min(tingkatKesulitan1.sedang(),
        Math.min(waktu1.mendadak(), jarak1.dekat()));
    xHarga[10] = Math.min(tingkatKesulitan1.sedang(),
        Math.min(waktu1.mendadak(), jarak1.sedang()));
    xHarga[11] = Math.min(tingkatKesulitan1.sedang(),
        Math.min(waktu1.mendadak(), jarak1.jauh()));
    xHarga[12] = Math.min(tingkatKesulitan1.sedang(),
        Math.min(waktu1.sedang(), jarak1.dekat()));
    xHarga[13] = Math.min(tingkatKesulitan1.sedang(),
        Math.min(waktu1.sedang(), jarak1.sedang()));
    xHarga[14] = Math.min(tingkatKesulitan1.sedang(),
        Math.min(waktu1.sedang(), jarak1.jauh()));
    xHarga[15] = Math.min(tingkatKesulitan1.sedang(),
        Math.min(waktu1.tdkmendadak(), jarak1.dekat()));
    xHarga[16] = Math.min(tingkatKesulitan1.sedang(),
        Math.min(waktu1.tdkmendadak(), jarak1.sedang()));
```

Kode Sumber 4.40 Implementasi Fungsi `hitungX()` (1)

```

xHarga[17] = Math.min(tingkatKesulitan1.sedang(),
    Math.min(waktul.tdkmendadak(), jarak1.jauh()));
xHarga[18] = Math.min(tingkatKesulitan1.sulit(),
    Math.min(waktul.mendadak(), jarak1.dekat()));
xHarga[19] = Math.min(tingkatKesulitan1.sulit(),
    Math.min(waktul.mendadak(), jarak1.sedang()));
xHarga[20] = Math.min(tingkatKesulitan1.sulit(),
    Math.min(waktul.mendadak(), jarak1.jauh()));
xHarga[21] = Math.min(tingkatKesulitan1.sulit(),
    Math.min(waktul.sedang(), jarak1.dekat()));
xHarga[22] = Math.min(tingkatKesulitan1.sulit(),
    Math.min(waktul.sedang(), jarak1.sedang()));
xHarga[23] = Math.min(tingkatKesulitan1.sulit(),
    Math.min(waktul.sedang(), jarak1.jauh()));
xHarga[24] = Math.min(tingkatKesulitan1.sulit(),
    Math.min(waktul.tdkmendadak(), jarak1.dekat()));
xHarga[25] = Math.min(tingkatKesulitan1.sulit(),
    Math.min(waktul.tdkmendadak(), jarak1.sedang()));
xHarga[26] = Math.min(tingkatKesulitan1.sulit(),
    Math.min(waktul.tdkmendadak(), jarak1.jauh()));
}

```

Kode Sumber 4.41 Implementasi Fungsi *hitungX* () (2)

Pada fungsi *hitungY* () nilai minimum hasil dari perhitungan pada fungsi *hitungX* () akan diolah kembali pada fungsi keanggotaan harga dengan cara memanggil domain kelas harga yang sesuai dengan penarikan kesimpulan pada masing-masing *rule*.

```

public static void hitungY(){
    yHarga[0] = Harga.murah(xHarga[0]);
    yHarga[1] = Harga.sedang(xHarga[1]);
    yHarga[2] = Harga.sedang(xHarga[2]);
    yHarga[3] = Harga.murah(xHarga[3]);
    yHarga[4] = Harga.sedang(xHarga[4]);
    yHarga[5] = Harga.sedang(xHarga[5]);
    yHarga[6] = Harga.murah(xHarga[6]);
    yHarga[7] = Harga.murah(xHarga[7]);
    yHarga[8] = Harga.sedang(xHarga[8]);
    yHarga[9] = Harga.sedang(xHarga[9]);
    yHarga[10] = Harga.sedang(xHarga[10]);
    yHarga[11] = Harga.mahal(xHarga[11]);
    yHarga[12] = Harga.sedang(xHarga[12]);
}

```

Kode Sumber 4.42 Implementasi Fungsi *hitungY* () (1)

```

yHarga[13] = Harga.sedang(xHarga[13]);
yHarga[14] = Harga.mahal(xHarga[14]);
yHarga[15] = Harga.murah(xHarga[15]);
yHarga[16] = Harga.sedang(xHarga[16]);
yHarga[17] = Harga.sedang(xHarga[17]);
yHarga[18] = Harga.sedang(xHarga[18]);
yHarga[19] = Harga.mahal(xHarga[19]);
yHarga[20] = Harga.mahal(xHarga[20]);
yHarga[21] = Harga.sedang(xHarga[21]);
yHarga[22] = Harga.mahal(xHarga[22]);
yHarga[23] = Harga.mahal(xHarga[23]);
yHarga[24] = Harga.sedang(xHarga[24]);
yHarga[25] = Harga.sedang(xHarga[25]);
yHarga[26] = Harga.mahal(xHarga[26]);
}

```

Kode Sumber 4.43 Implementasi Fungsi `hitungY()` (2)

Tidak jauh beda dengan sebelumnya, pada proses fuzzyfikasi dalam perhitungan estimasi harga terdapat 9 aturan (*rule*) yang dibangun ke dalam sistem. Jumlah aturan tersebut didapatkan dari hasil perhitungan $\text{Jumlah_Domain}^{\text{Jumlah_Variabel}}$. Proses ini akan dilakukan terhadap masukan yang diberikan *user* berupa tingkat kesulitan dan waktu pada fungsi `hitungX()`.

Pada fungsi ini proses dimulai dengan mengambil masukan dan memanggil kelas masukan yang berkaitan untuk mendapatkan nilai *fuzzy* atau derajat keanggotaannya. Kemudian akan diambil nilai terendah dari kedua masukan tersebut menggunakan metode MIN dengan mengambil nilai terendah antara tingkat kesulitan dan waktu. Setelah didapatkan nilai minimum dari kedua variabel pada masing-masing *rule* tersebut, kemudian dilakukan penarikan kesimpulan pada fungsi `hitungY()`.

```

public static void hitungX(TingkatKesulitan
tingkatKesulitan1, Waktu waktul){
    xHarga[0] = Math.min(tingkatKesulitan1.mudah(),
        waktul.tdkmendadak());
    xHarga[1] = Math.min(tingkatKesulitan1.mudah(),
        waktul.sedang());
    xHarga[2] = Math.min(tingkatKesulitan1.mudah(),
        waktul.mendadak());
    xHarga[3] = Math.min(tingkatKesulitan1.sedang(),
        waktul.tdkmendadak());
    xHarga[4] = Math.min(tingkatKesulitan1.sedang(),
        waktul.sedang());
    xHarga[5] = Math.min(tingkatKesulitan1.sedang(),
        waktul.mendadak());
    xHarga[6] = Math.min(tingkatKesulitan1.sulit(),
        waktul.tdkmendadak());
    xHarga[7] = Math.min(tingkatKesulitan1.sulit(),
        waktul.sedang());
    xHarga[8] = Math.min(tingkatKesulitan1.sulit(),
        waktul.mendadak());
}

```

Kode Sumber 4.44 Implementasi `hitungX()` Estimasi Harga

Pada fungsi `hitungY()` nilai minimum hasil dari perhitungan pada fungsi `hitungX()` akan diolah kembali pada fungsi keanggotaan harga dengan cara memanggil domain kelas harga yang sesuai dengan penarikan kesimpulan pada masing-masing *rule*.

```

public static void hitungY(){
    yHarga[0] = Harga.murah(xHarga[0]);
    yHarga[1] = Harga.murah(xHarga[1]);
    yHarga[2] = Harga.sedang(xHarga[2]);
    yHarga[3] = Harga.murah(xHarga[3]);
    yHarga[4] = Harga.sedang(xHarga[4]);
    yHarga[5] = Harga.sedang(xHarga[5]);
    yHarga[6] = Harga.sedang(xHarga[6]);
    yHarga[7] = Harga.sedang(xHarga[7]);
    yHarga[8] = Harga.mahal(xHarga[8]);
}

```

Kode Sumber 4.45 Implementasi `hitungY()` Estimasi Harga

4.3.4 Implementasi *Defuzzyfication*

Proses *defuzzyfication* pada perhitungan harga dan estimasi harga akan dilakukan pada fungsi yang sama yaitu `defuzzyfikasi()`. Pada fungsi ini akan dilakukan penjumlahan hasil dari perhitungan fungsi `hitungX()` berupa nilai minimum dari derajat keanggotaan yang dikalikan dengan nilai *crisp* hasil perhitungan fungsi `hitungY()`. Kemudian hasil penjumlahan dari perhitungan tersebut akan dibagi dengan jumlah nilai dari nilai minimum derajat keanggotaan pada fungsi `hitungX()`.

```
public static double defuzzyfikasi(){
    double atas = 0, bawah = 0;
    for(int i = 0 ; i < 27 ; i++){
        atas += (xHarga[i] * yHarga[i]);
        bawah += xHarga[i];
    }
    System.out.println(atas/bawah);
    return (atas/bawah);
}
```

Kode Sumber 4.46 Implementasi Fungsi `defuzzyfikasi()`

4.3.5 Implementasi Perhitungan Harga Akhir

Proses perhitungan harga akan dimulai dengan cara memanggil fungsi `hitungX()` dengan parameter tingkat kesulitan, waktu dan jarak. Setelah itu akan dipanggil fungsi `hitungY()` untuk melakukan penarikan kesimpulan dari hasil *fuzzyfication* pada fungsi sebelumnya. Hasil dari fungsi tersebut akan di-defuzzyfikasi untuk mendapatkan harga awal.

Harga awal merupakan harga dari transaksi dengan durasi 1,5 jam atau 90 menit sehingga harus dilakukan perhitungan untuk mendapatkan harga yang sesuai dengan durasi yang di masukan oleh *user*. Setelah itu jarak dari transaksi tersebut akan dicocokkan pada 3 zona daerah untuk mengetahui tambahan biaya transportasi yang akan ditambahkan pada harga awal. Harga akhir merupakan harga awal hasil defuzzyfikasi yang telah disesuaikan dengan durasi dan ditambah dengan biaya transportasi berdasarkan letak jarak pada 3 zona yang telah ditentukan sebelumnya.

```

Fuzzy fuzzy = new Fuzzy();
fuzzy.hitungX(tingkatKesulitan, waktu, jarak);
fuzzy.hitungY();

hargaawal = fuzzy.defuzzyfikasi()*dataMurid.getDurasi()/90;

if(jarak.getjarak() > 5000 && jarak.getjarak() <= 10000){
    hargaawal = hargaawal + 5000;
}
else if(jarak.getjarak() > 10000 && jarak.getjarak() <= 15000){
    hargaawal = hargaawal + 10000;
}
else if (jarak.getjarak() > 15000 ){
    hargaawal = hargaawal + 20000;
}

```

Kode Sumber 4.47 Implementasi Perhitungan Harga Akhir

4.4 Implementasi Website Statistik Proses Bisnis

Website statistik proses bisnis ini diimplementasikan untuk memantau proses bisnis yang telah terjadi didalam aplikasi “*Finding Tutor*”. Pengguna baik admin maupun *developer* dapat memantau transaksi secara keseluruhan seperti pendaftaran tutor maupun murid, daftar tutor maupun murid terbaik, transaksi yang telah terjadi dan laporan pemasukan hasil transaksi. Semua fitur dilengkapi dengan *filter* yang memungkinkan pengguna untuk mengetahui laporan sesuai dengan waktu yang diinginkan. Selain juga terdapat fitur cetak atau *download* dalam beberapa bentuk *file*.

4.4.1 Implementasi Halaman Utama (*Dashboard*)

Pada halaman utama (*Dashboard*) terdapat beberapa fitur seperti jumlah tutor dan murid yang mendaftar hari ini, jumlah transaksi yang telah berhasil dilakukan hari ini dan jumlah pemasukan dari transaksi yang telah berhasil hari ini. Selain fitur-fitur utama tersebut, terdapat fitur tambahan berupa representasi grafik transaksi pada tiap-tiap bulan dan tahunnya. Pada fitur pendaftaran tutor hari ini, jumlah didapatkan dari *query* COUNT pada tabel *user* dimana tanggal daftar *user* sama dengan tanggal sistem pada hari tersebut dan jenis *user* sama dengan pentutor.

Controllers

```
$this->load->model('mtutor');
$data['query1'] = $this->mtutor->TutorHariIni();
```

Model

```
function TutorHariIni(){
    $query = "SELECT COUNT(*) as jumlah
              FROM user
              WHERE date(tanggal_daftar) = CURDATE()
              AND jenis_user = 'Pentutor'";
    $hasil = $this->db->query($query);
    return $hasil->result();
}
```

Kode Sumber 4.48 Implementasi Pendaftar Tutor Hari Ini

Pada fitur pendaftar murid hari ini, jumlah didapatkan dari *query* COUNT pada tabel *user* dimana tanggal daftar user sama dengan tanggal sistem pada hari tersebut dan jenis *user* adalah murid.

Controllers

```
$this->load->model('mmurid');
$data['query2'] = $this->mmurid->MuridHariIni();
```

Model

```
function MuridHariIni(){
    $query = "SELECT COUNT(*) as jumlah FROM user
              WHERE date(tanggal_daftar) = CURDATE()
              AND jenis_user = 'Murid'";
    $hasil = $this->db->query($query);
    return $hasil->result();
}
```

Kode Sumber 4.49 Implementasi Pendaftar Murid Hari Ini

Pada fitur transaksi hari ini, jumlah transaksi didapatkan dari *query* COUNT pada tabel transaksi dimana tanggal transaksi sama dengan tanggal sistem pada hari tersebut.

Controllers

```
$this->load->model('mtransaksi');
$data['query3'] =
    $this->mtransaksi->TransaksiHariIni();
```

Model

```
function TransaksiHariIni(){
    $query ="SELECT COUNT(*) as jumlah FROM transaksi
        WHERE date(tanggal_transaksi) = CURDATE()";
    $hasil = $this->db->query($query);
    return $hasil->result();
}
```

Kode Sumber 4.50 Implementasi Transaksi Hari Ini

Pada fitur pemasukan hari ini, jumlah pemasukan transaksi didapatkan dari *query* SUM kolom biayatutor_pecarian pada tabel pencarian_tutor dimana tanggal transaksi sama dengan tanggal sistem pada hari tersebut.

Controllers

```
$this->load->model('mtransaksi');
$data['query4'] =
    $this->mtransaksi->PemasukanHariIni();
```

Model

```
function PemasukanHariIni(){
    $query = "SELECT SUM(biayatutor_pecarian)
        as jumlah
        FROM pencarian_tutor
        WHERE date(tanggal_transaksi) = CURDATE()";
    $hasil = $this->db->query($query);
    return $hasil->result();
}
```

Kode Sumber 4.51 Implementasi Jumlah Pemasukan Hari Ini

Pada fitur grafik transaksi per-bulan dan per-tahun nilai transaksi didapatkan dan dikelompokkan dari *query* pada tabel transaksi berdasarkan bulan dan tahun. Representasi dari hasil kedua *query* tersebut dibangun kedalam 2 buah grafik yang diimplementasikan menggunakan *javascript*.

Controllers

```
$this->load->model('mchart');
$data['bulan'] = $this->mchart->chartBulan();
```

Model

```
function chartBulan() {
    $query = "SELECT monthname(tanggal_transaksi)
              as month, count(id_transaksi) as num
              FROM transaksi
              GROUP BY (extract(month FROM tanggal_transaksi))";
    $hasil = $this->db->query($query);
    return $hasil->result();
}
```

Kode Sumber 4.52 Implementasi Grafik Transaksi Per-Bulan

Controllers

```
$this->load->model('mchart');
$data['bulan'] = $this->mchart->chartBulan();
```

Model

```
function chartTahun() {
    $query = "SELECT year(tanggal_transaksi) as year,
                    count(id_transaksi) as num
                    FROM transaksi
                    GROUP BY (extract(year FROM tanggal_transaksi))";
    $hasil = $this->db->query($query);
    return $hasil->result();
}
```

Kode Sumber 4.53 Implementasi Grafik Transaksi Per-Tahun

4.4.2 Implementasi Halaman Data Pendaftar Tutor

Halaman data pendaftar tutor dapat dimanfaatkan untuk melihat rekap data tutor yang telah terdaftar pada sistem pada periode bulan dan tahun tertentu. Pada halaman ini data tutor yang akan ditampilkan yaitu id user, tanggal daftar, nama, alamat, jenis kelamin, usia, telepon, email dan status sebagai pentutor. Pada halaman ini terdapat fitur pencarian dan sorting data tutor yang ada pada sistem. Selain itu, *user* juga dapat *men-download* rekap data sebagai *file CSV*, *excel* maupun *pdf*.

Controllers

```

-----
public function data_tutor(){
    $bulan = $this->input->get('bulan');
    $tahun = $this->input->get('tahun');
    $this->load->model('mtutor');
    $data['query'] = $this->mtutor->selectAll($bulan, $tahun);
    $this->load->view('data_tutor', $data);
}

```

Model

```

-----
function selectAll($bulan, $tahun) {
    if(is_null($bulan) && is_null($tahun)){
        $query = "SELECT * FROM `user`
                WHERE `jenis_user` = 'Pentutor'
                AND MONTH(tanggal_daftar) = '01'
                AND YEAR(tanggal_daftar) = '2017'";
        $hasil = $this->db->query($query);
        return $hasil->result();
    }else{
        $query = "SELECT * FROM `user`
                WHERE `jenis_user` = 'Pentutor'
                AND MONTH(tanggal_daftar) = $bulan
                AND YEAR(tanggal_daftar) = $tahun";
        $hasil = $this->db->query($query);
        return $hasil->result();
    }
}

```

Kode Sumber 4.54 Implementasi Data Pendaftar Tutor

4.4.3 Implementasi Halaman Data Tutor Terbaik

Halaman data tutor terbaik dapat dimanfaatkan untuk melihat 5 tutor terbaik yang ditentukan dari banyaknya transaksi yang telah dilakukan pada bulan tertentu. Pada halaman ini data tutor terbaik yang akan ditampilkan yaitu nama tutor, alamat, jenis kelamin, usia, nomor telepon, email dan jumlah transaksi. Pada halaman ini terdapat fitur pencarian data tutor terbaik. Selain itu, *user* juga dapat *men-download* maupun mencetak rekap data sebagai *file CSV*, *excel* maupun *pdf*.

Controllers

```
public function data tutor(){
    $bulan = $this->input->get('bulan');
    $tahun = $this->input->get('tahun');
    $this->load->model('mtutor');
    $data['query'] = $this->mtutor->selectAll($bulan, $tahun);
    $this->load->view('data_tutor', $data);
}
```

Kode Sumber 4.55 Implementasi *Controllers* Data Tutor Terbaik

Model

```
function selectAll($bulan, $tahun) {
    if(is_null($bulan) && is_null($tahun)){
        $query = "SELECT `username_tutor`, user.`alamat_user`,
            user.`jeniskelamin_user`, user.`usia_user`,
            user.`telp_user`, user.`email_user`,
            COUNT(*) AS jumlah
            FROM history_transaksi INNER JOIN user
            WHERE history_transaksi.`username_tutor`
                = user.`username_user`
                AND MONTH(tanggal) = '01'
                AND YEAR(tanggal) = '2017'
                GROUP BY username_tutor DESC LIMIT 5";
        $hasil = $this->db->query($query);
        return $hasil->result();
    }
}
```

Kode Sumber 4.56 Implementasi *Model* Data Tutor Terbaik (1)

```

else{
    $query = "SELECT `username_tutor`, user.`alamat_user`,
                user.`jeniskelamin_user`, user.`usia_user`,
                user.`telp_user`, user.`email_user`,
                COUNT(*) AS jumlah
            FROM history_transaksi INNER JOIN user
            WHERE history_transaksi.`username_tutor`
                    = user.`username_user`
                    AND MONTH(tanggal) = $bulan
                    AND YEAR(tanggal) = $tahun
                    GROUP BY username_tutor DESC LIMIT 5";
    $hasil = $this->db->query($query);
    return $hasil->result();
}
}

```

Kode Sumber 4.57 Implementasi *Model* Data Tutor Terbaik (2)

4.4.4 Implementasi Halaman Data Pendaftar Murid

Halaman data pendaftar murid dapat dimanfaatkan untuk melihat rekap data murid yang telah terdaftar pada sistem pada periode bulan dan tahun tertentu. Pada halaman ini data murid yang akan ditampilkan yaitu id user, tanggal daftar, nama, alamat, jenis kelamin, usia, telepon, email dan status sebagai murid. Pada halaman ini terdapat fitur pencarian dan sorting data murid yang ada pada sistem. Selain itu, *user* juga dapat men-*download* rekap data sebagai *file* CSV, *excel* maupun *pdf*.

```

Controllers
-----
public function data_murid(){
    $bulan = $this->input->get('bulan');
    $tahun = $this->input->get('tahun');
    $this->load->model('mmurid');
    $data['query'] = $this->mmurid->selectAll($bulan, $tahun);
    $this->load->view('data_murid', $data);
}

```

Kode Sumber 4.58 Implementasi *Controllers* Data Pendaftar Murid

```

Model
-----
function selectAll($bulan, $tahun) {
    if(is_null($bulan) && is_null($tahun)){
        $query = "SELECT * FROM `user`
                WHERE `jenis_user` = 'Murid'
                AND MONTH(tanggal_daftar) = '01'
                AND YEAR(tanggal_daftar) = '2017'";
        $hasil = $this->db->query($query);
        return $hasil->result();
    }else{
        $query = "SELECT * FROM `user`
                WHERE `jenis_user` = 'Murid'
                AND MONTH(tanggal_daftar) = $bulan
                AND YEAR(tanggal_daftar) = $tahun";
        $hasil = $this->db->query($query);
        return $hasil->result();
    }
}

```

Kode Sumber 4.59 Implementasi Model Data Pendaftar Murid

4.4.5 Implementasi Halaman Data Murid Terbaik

Halaman data murid terbaik dapat dimanfaatkan untuk melihat 5 murid terbaik yang ditentukan dari banyaknya transaksi yang telah dilakukan pada bulan tertentu. Pada halaman ini data murid terbaik yang akan ditampilkan yaitu nama murid, alamat, jenis kelamin, usia, nomor telepon, email dan jumlah transaksi. Pada halaman ini terdapat fitur pencarian data murid terbaik. Selain itu, *user* juga dapat *men-download* maupun mencetak rekap data sebagai *file CSV*, *excel* maupun *pdf*.

Controllers

```

-----
public function murid_terbaik(){
    $bulan = $this->input->get('bulan');
    $tahun = $this->input->get('tahun');
    $this->load->model('mbestmurid');
    $data['query'] = $this->mbestmurid->selectAll($bulan, $tahun);
    $this->load->view('murid_terbaik', $data);
}

```

Model

```

function selectAll($bulan, $tahun){
    if(is_null($bulan) && is_null($tahun)){
        $query = "SELECT  user.`username_user`,
                        user.`alamat_user`,
                        user.`jeniskelamin_user`, user.`usia_user`,
                        user.`telp_user`, user.`email_user`, S
                    FROM USER INNER JOIN
                    ( SELECT `nameuser_pencarian`, COUNT(*) AS S
                      FROM pencarian_tutor
                      INNER JOIN history_transaksi
                      WHERE MONTH(tanggal) = '01'
                      AND YEAR(tanggal) = '2017'
                      AND `id_pencarian`=
                          history_transaksi.`id_pencariantutor`
                      GROUP BY nameuser_pencarian) AS coba
                    ON user.nama_user = coba.nameuser_pencarian
                    ORDER BY S DESC LIMIT 5";
        $hasil = $this->db->query($query);
        return $hasil->result();
    }else{
        $query = "SELECT  user.`username_user`,
                        user.`alamat_user`, user.`jeniskelamin_user`,
                        user.`usia_user`, user.`telp_user`,
                        user.`email user`, S
                    FROM USER INNER JOIN
                    ( SELECT `nameuser_pencarian`, COUNT(*) AS S
                      FROM pencarian_tutor
                      INNER JOIN history_transaksi
                      WHERE MONTH(tanggal) = $bulan
                      AND YEAR(tanggal) = $tahun
                      AND `id_pencarian`=
                          history_transaksi.`id_pencariantutor`
                      GROUP BY nameuser_pencarian) AS coba
                    ON user.nama_user = coba.nameuser_pencarian
                    ORDER BY S DESC LIMIT 5";
        $hasil = $this->db->query($query);
        return $hasil->result();
    }
}

```

Kode Sumber 4.60 Implementasi Data Murid Terbaik

4.4.6 Implementasi Data Transaksi

Halaman data laporan transaksi dapat dimanfaatkan untuk melihat rekap transaksi yang telah berhasil dilakukan. Pada halaman ini data transaksi yang akan ditampilkan yaitu id transaksi, tanggal transaksi, nama tutor, nama murid, kelas, pelajaran, alamat, durasi dan biaya. Pada halaman ini terdapat fitur pencarian yang ada pada sistem. Selain itu, *user* juga dapat *men-download* rekap data sebagai *file CSV*, *excel* maupun *pdf*.

Controllers

```
public function transaksi(){
    $bulan = $this->input->get('bulan');
    $tahun = $this->input->get('tahun');
    $this->load->model('mtransaksi');
    $data['query'] = $this->mtransaksi->selectAll($bulan, $tahun);
    $this->load->view('transaksi', $data);
}
```

Model

```
function selectAll($bulan, $tahun){
    if(is_null($bulan) && is_null($tahun)){
        $query = "SELECT `id_transaksi`, `biayatutor_pencarian`
        transaksi.`id_pencariantutor`, history_transaksi.`tanggal`,
        pencarian_tutor.`id_pencarian`, `pelajaran_pencarian`,
        history_transaksi.`id_pencariantutor`,
        history_transaksi.`username_tutor`, `nameuser_pencarian`
        `kelas_pencarian`, `alamat_pencarian`, `durasi_pencarian`,
        FROM transaksi inner join pencarian_tutor
        INNER JOIN history_transaksi
        WHERE transaksi.`id_pencariantutor` =
        pencarian_tutor.`id_pencarian`
        AND pencarian_tutor.`id_pencarian` =
        history_transaksi.`id_pencariantutor`
        AND MONTH(tanggal) = '01'
        AND YEAR(tanggal) = '2017'";
        $hasil = $this->db->query($query);
        return $hasil->result();
    }
```

Kode Sumber 4.61 Implementasi Data Transaksi (1)

```

}else{
    $query = "SELECT `id_transaksi`, `biayatutor_pencarian`,
    transaksi.`id_pencariantutor`,history_transaksi.`tanggal`,
    pencarian_tutor.`id_pencarian`, `pelajaran_pencarian`,
    history_transaksi.`id_pencariantutor`,
    history_transaksi.`username_tutor`, `nameuser_pencarian`,
    `kelas_pencarian`, `alamat_pencarian`,`durasi_pencarian`
    FROM transaksi inner join pencarian_tutor
    INNER JOIN history_transaksi
    WHERE transaksi.`id_pencariantutor` =
        pencarian_tutor.`id_pencarian`
    AND pencarian_tutor.`id_pencarian` =
        history_transaksi.`id_pencariantutor`
    AND MONTH(tanggal) = $bulan
    AND YEAR(tanggal) = $tahun";
    $hasil = $this->db->query($query);
    return $hasil->result();
}
}

```

Kode Sumber 4.62 Implementasi Data Transaksi (2)

BAB V

UJI COBA DAN EVALUASI

Dalam bab ini dibahas mengenai hasil uji coba sistem yang telah dirancang dan dibuat. Uji coba dilakukan untuk mengetahui kinerja sistem dengan lingkungan uji coba yang telah ditentukan.

5.1 Lingkungan Uji Coba

Lingkungan uji coba pada tugas akhir ini adalah sebuah emulator pada laptop (*notebook*). Spesifikasi *notebook* dari sisi perangkat keras adalah memiliki prosesor Intel Core i7 3537U dengan kecepatan 2,00 GHz dan memori sebesar 8,00 GB. *Notebook* yang digunakan memiliki sistem operasi Windows 8.1.

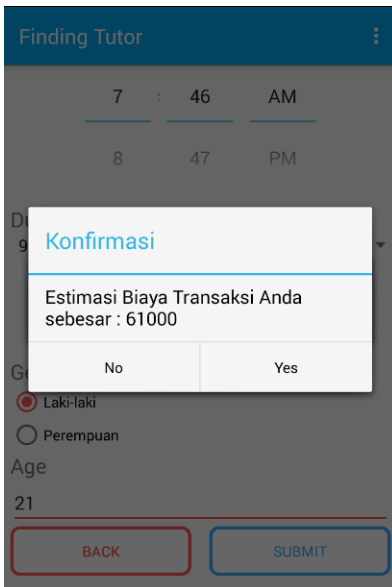
Pada sisi perangkat lunak, uji coba pada tugas akhir ini dilakukan dengan menggunakan aplikasi berbasis Android yaitu *Finding-Tutor*. Aplikasi ini merupakan aplikasi pencarian tutor yang dapat memudahkan *user* untuk mendapatkan tutor. Setelah *user* mengisi formulir pencarian tutor, aplikasi ini akan menampilkan estimasi harga dan harga akhir dari hasil perhitungan harga menggunakan metode *Fuzzy Tsukamoto* yang telah diimplementasikan kedalam sistem. Uji coba akan dilakukan pada beberapa skenario masukan yang diberikan *user*.

5.2 Data Uji Coba

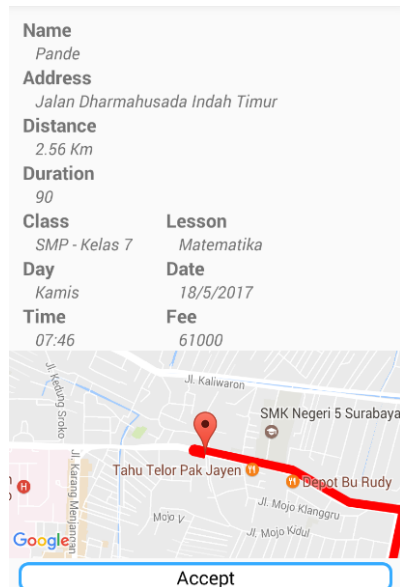
Data yang akan digunakan untuk uji coba implementasi perhitungan harga tutor menggunakan metode *Fuzzy Tsukamoto* adalah masukan dari beberapa *user* berupa tingkat kesulitan, tanggal transaksi dan alamat murid serta alamat 1 *user* yang berperan sebagai tutor. Aktor pada uji coba sebanyak 6 orang diantaranya lain 5 orang sebagai pencari tutor dengan alamat yang berbeda dan 1 orang sebagai tutor. Latar belakang dari aktor yang melakukan uji coba merupakan mahasiswa dan orang tua murid yang membutuhkan bantuan tutor dan 1 aktor mahasiswa yang memiliki latar belakang sebagai guru les privat pada area Surabaya. Skenario masukan dapat diamati pada **Tabel 5.1 dan Tabel 5.2**.

5.3 Uji Coba

Uji coba dilakukan untuk mengetahui apakah seluruh fitur pada aplikasi *mobile* maupun *website* dapat berjalan dengan baik serta harga yang dihasilkan dari perhitungan sesuai dengan kategori harga dan aturan yang telah ditentukan pada sistem sebelumnya. Dari hasil uji coba pada kedua perhitungan yang telah dilakukan di dapatkan,

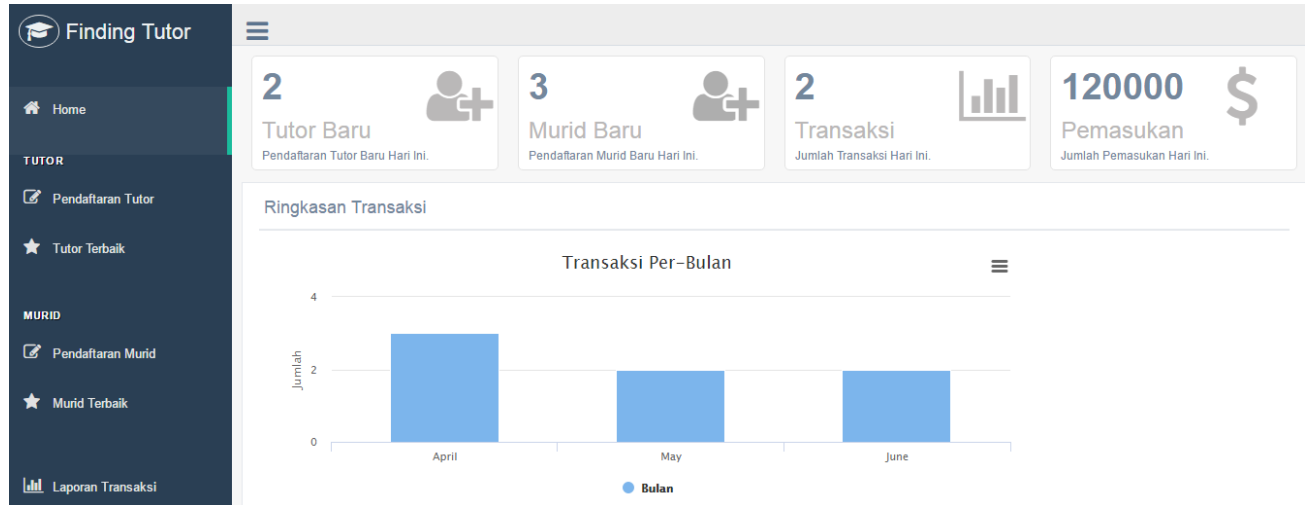


(a)

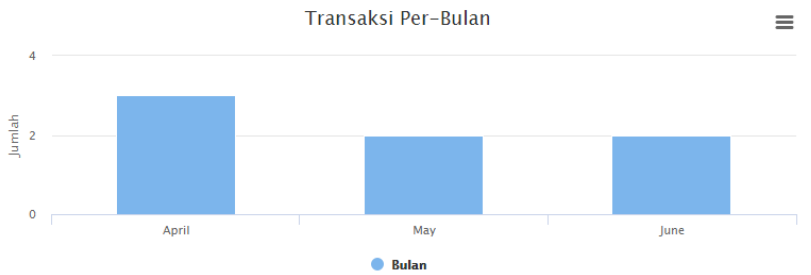


(b)

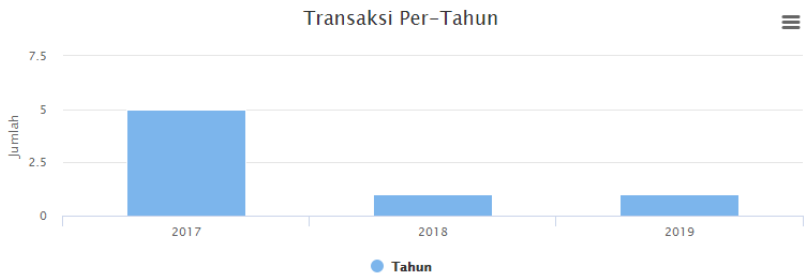
Gambar 5.1 Tampilan (a) Estimasi Harga (b) Harga Akhir



Gambar 5.2 Tampilan Halaman Utama (Dashboard)



(a)



(b)

Gambar 5.3 Tampilan Representasi Grafik (a) Bulan (b) Tahun

Pendaftaran Tutor

Rekap Pendaftaran Tutor

Bulan

Tahun

Januari

2017

Tampilkan

Copy

CSV

Excel

Print

Search:

ID USER	TANGGAL DAFTAR	NAMA	ALAMAT	JENIS KELAMIN	USIA	TELEPON	EMAIL	JENIS
8	2017-05-03 22:20:06	Riska Adhita	Teknik Informatika ITS	Laki-laki	21	085640794801	risk.adhita@gmail.com	Pentutor
15	2017-05-21 22:20:06	Laurensius Adi	Dharmahusada	Laki-laki	20	23908664846	lino@gmail.com	Pentutor
16	2017-05-21 22:20:06	Kinasih Nur Azizah	Gubeng	Perempuan	21	081357391436	alfonza@gmail.com	Pentutor

Showing 1 to 3 of 3 entries

Previous1Next

Gambar 5.4 Tampilan Halaman Data Pendaftar Tutor

Tutor Terbaik

Rekap 5 Top Tutor of The Month

^ x

Bulan

Tahun

Januari

2017

Tampilkan

Copy

CSV

Excel

Print

Search:

NAMA TUTOR	ALAMAT	JENIS KELAMIN	USIA	NOMOR TELPON	EMAIL	JUMLAH TRANSAKSI
lino	Royal Plaza	Laki-laki	21	085640794802	reizkardhitio@gmail.com	1
r_adhita	Teknik Informatika ITS	Laki-laki	21	085640794801	risk.adhita@gmail.com	1

Showing 1 to 2 of 2 entries

Previous 1 Next

Gambar 5.5 Tampilan Halaman Data Tutor Terbaik

Pendaftaran Murid

Rekap Pendaftaran Murid (/Hari /Bulan /Tahun)

^ x

Bulan Tahun

Januari 2017

Tampilkan

Copy CSV Excel Print

Search:

ID USER	TANGGAL DAFTAR	NAMA	ALAMAT	JENIS KELAMIN	USIA	TELEPON	EMAIL	JENIS
9	2017-05-21 22:20:06	Luwandino Wismar	Royal Plaza	Laki-laki	21	085640794802	reizkardhitio@gmail.com	Murid
17	2017-05-21 22:20:06	Naufal Fachri	Tunjungan Plaza	Laki-laki	21	081965879432	naufal@gmail.com	Murid
25	2017-05-21 22:20:06	Pande Nyoman	Keputih	Laki-laki	22	081234567890	pande@gmail.com	Murid

Showing 1 to 3 of 3 entries

Previous 1 Next

Gambar 5.6 Tampilan Halaman Data Pendaftar Murid

Murid Terbaik

Rekap 5 Top Murid of The Month



Bulan

Tahun

Januari

2017

Tampilkan

Copy

CSV

Excel

Print

Search:

NAMA MURID	ALAMAT	JENIS KELAMIN	USIA	NOMOR TELPON	EMAIL	JUMLAH TRANSAKSI
pande	Keputih	Laki-laki	22	081234567890	pande@gmail.com	1

Showing 1 to 1 of 1 entries

Previous 1 Next

Gambar 5.7 Tampilan Halaman Data Murid Terbaik

Transaksi

Rekap Transaksi (/Hari /Bulan /Tahun)

Bulan

Tahun

Tampilkan

Copy

CSV

Excel

Print

Search:

ID	TANGGAL TRANSAKSI	NAMA TUTOR	NAMA MURID	KELAS	PELAJARAN	ALAMAT	DURASI	BIAYA
3	2017-04-09 22:19:31	r_adhita	Naufal	UMUM	C++	Galaxy Mall	60	66000
5	2017-04-03 22:19:31	lino	Pande Nyoman	SD - Kelas 5	Kimia	Galaxy Mall	90	60000

Showing 1 to 2 of 2 entries

Previous1Next

Gambar 5.8 Tampilan Halaman Data Transaksi

Tabel 5.1 Tabel Uji Coba UC.001

ID	UJ.UC.001
Referensi Kasus Penggunaan	UC. 001
Nama	Mendapatkan Estimasi Harga
Tujuan Pengujian	Menguji kemampuan sistem dalam perhitungan estimasi harga dan menampilkan data <i>pop up</i> pada halaman pencarian tutor
Skenario	Pengguna memasukkan kriteria pencarian tutor
Kondisi Awal	Pengguna berada pada halaman utama murid
Data Uji	Kriteria pencarian tutor : 1. Kelas : SD Kelas 3 2. Mata Pelajaran : Matematika 3. Alamat : Jalan Manyar Kertoarjo 4. Tanggal : 04/06/2017 5. Waktu : 17.00 6. Durasi : 90 Menit 7. Gender : Perempuan 8. Max Umur : 28 Tahun
Langkah Pengujian	1. Aktor memilih menu pencarian murid 2. Memasukkan kriteria pencarian murid 3. Memilih detail murid yang diinginkan
Hasil yang diharapkan	Harga akhir berhasil ditampilkan pada halaman detail murid
Hasil yang didapat	Harga akhir berhasil ditampilkan pada halaman detail murid

Hasil dari uji coba pada tabel UJ.UC.001 dapat dilihat pada **Tabel 5.4** nomor 13 dimana didapatkan hasil estimasi harga sebesar Rp 61.000 yang dinyatakan memiliki nilai yang sesuai berdasarkan dengan aturan yang telah dibangun.

Tabel 5.2 Tabel Uji Coba UC. 002

ID	UJ.UC.002
Referensi Kasus Penggunaan	UC. 002
Nama	Mendapatkan Harga Akhir
Tujuan Pengujian	Menguji kemampuan sistem dalam perhitungan harga akhir dan menampilkan data pada halaman detil murid
Skenario	Pengguna memasukkan kriteria pencarian murid
Kondisi Awal	Pengguna berada pada halaman utama tutor
Data Uji	Kriteria pencarian murid : Semua kriteria
Langkah Pengujian	<ol style="list-style-type: none"> 1. Aktor memilih menu pencarian murid 2. Memasukkan kriteria pencarian murid 3. Memilih detil murid yang diinginkan
Hasil yang diharapkan	Harga akhir berhasil ditampilkan pada halaman detil murid
Hasil yang didapat	Harga akhir berhasil ditampilkan pada halaman detil murid

Hasil dari uji coba pada tabel UJ.UC.002 dapat dilihat pada **Tabel 5.5** nomor 6 dimana didapatkan hasil harga akhir lebih murah daripada estimasi harga yaitu sebesar Rp 53.000 dan dinyatakan memiliki nilai yang sesuai dengan aturan yang telah dibangun.

Tabel 5.3 Kategori Harga Durasi 90 Menit

Harga	Keterangan
Rp 35.000 – Rp 60.000	Murah
Rp 45.000 – Rp 80.000	Sedang
Rp 80.000 – Rp 100.000	Mahal

Tabel 5.4 Hasil Uji Coba Perhitungan Estimasi Harga

No.	Tingkat Kesulitan	Tanggal Transaksi	Harga	Hasil
1	SD - 1	09/06/2017	Murah	46.000
2	SD - 2	10/06/2017	Murah	46.000
3	SD - 3	12/06/2017	Murah	46.000
4	SD - 4	13/06/2017	Murah	46.000
5	SD - 5	15/06/2017	Murah	53.000
6	SD - 6	15/06/2017	Murah	46.000
7	SD - 1	05/06/2017	Murah	53.000
8	SD - 3	06/06/2017	Murah	46.000
9	SD - 4	07/06/2017	Murah	46.000
10	SD - 5	08/06/2017	Murah	53.000
11	SD - 6	09/06/2017	Murah	46.000
12	SD - 1	03/06/2017	Sedang	61.000
13	SD - 3	04/06/2017	Sedang	61.000
14	SD - 4	05/06/2017	Sedang	53.000
15	SD - 5	06/06/2017	Sedang	53.000
16	SD - 6	07/06/2017	Sedang	61.000
17	SD - 4	09/06/2017	Murah	46.000
18	SD - 5	10/06/2017	Murah	53.000
19	SD - 6	12/06/2017	Murah	46.000
20	SMP - 7	13/06/2017	Murah	46.000
21	SMP - 8	15/06/2017	Murah	46.000
22	SMP - 9	16/06/2017	Murah	46.000
23	SMP - 8	08/06/2017	Sedang	61.000
24	SMP - 9	09/06/2017	Sedang	61.000
25	SD - 4	03/06/2017	Sedang	61.000
26	SD - 5	04/06/2017	Sedang	53.000

27	SD - 6	05/06/2017	Sedang	61.000
28	SMP - 8	06/06/2017	Sedang	53.000
29	SMP - 9	07/06/2017	Sedang	61.000
30	SMP - 9	09/06/2017	Sedang	61.000
31	SMA - 10	10/06/2017	Sedang	50.000
32	SMA - 11	11/06/2017	Sedang	56.000
33	SMA - 12	12/06/2017	Sedang	61.000
34	UMUM - 13	14/06/2017	Sedang	61.000
35	SMP - 9	05/06/2017	Mahal	61.000
36	SMA - 10	06/06/2017	Mahal	67.000
37	SMA - 11	07/06/2017	Mahal	56.000
38	SMA - 12	08/06/2017	Mahal	61.000
39	UMUM - 13	09/06/2017	Mahal	61.000
40	SMP - 9	03/06/2017	Mahal	61.000
41	SMA - 10	04/06/2017	Mahal	84.000
42	SMA - 11	05/06/2017	Mahal	87.000
43	SMA - 12	06/06/2017	Mahal	69.000
44	UMUM - 13	07/06/2017	Mahal	61.000

Tabel 5.5 Hasil Uji Coba Perhitungan Harga Akhir

No	Tingkat Kesulitan	Waktu	Jarak	Harga	Hasil
1	SD - 1	03/06/2017	Jl. Raya Mulyosari	Murah	46.000
2	SD - 3	05/06/2017	Jl. Galaksi Klampis Asri Utara X	Murah	51.000
3	SD - 4	06/06/2017	Jl. Manyar Kertoarjo	Murah	53.000
4	SD - 5	07/06/2017	Jl. Raya Menur	Murah	53.000
5	SD - 1	05/06/2017	Jl. Galaksi Klampis Asri Utara X	Sedang	52.000
6	SD - 3	04/06/2017	Jl. Manyar Kertoarjo	Sedang	53.000
7	SD - 4	05/06/2017	Jl. Raya Menur	Sedang	60.000
8	SD - 5	06/06/2017	Jl. Kedung Cowek	Sedang	74.000
9	SD - 1	07/06/2017	Jl. Kedung Cowek	Sedang	61.000
10	SD - 3	06/06/2017	Taman Bungkul	Sedang	63.000
11	SD - 4	03/06/2017	Jl. Taman Mayangkara	Sedang	71.000
12	SD - 5	02/06/2017	Wonokromo	Sedang	79.000
13	SD - 2	05/06/2017	Jl. Raya Mulyosari	Murah	46.000
14	SD - 4	07/06/2017	Jl. Galaksi Klampis Asri Utara X	Murah	51.000

15	SD - 5	07/06/2017	Jl. Manyar Kertoarjo	Murah	53.000
16	SD - 6	08/06/2017	Jl. Raya Menur	Murah	59.000
17	SD - 2	05/06/2017	Jl. Galaksi Klampis Asri Utara X	Sedang	51.000
18	SD - 4	06/06/2017	Jl. Manyar Kertoarjo	Sedang	53.000
19	SD - 5	07/06/2017	Jl. Raya Menur	Sedang	53.000
20	SD - 6	09/06/2017	Jl. Kedung Cowek	Sedang	93.000
21	SD - 2	05/06/2017	Jl. Kedung Cowek	Sedang	61.000
22	SD - 4	06/06/2017	Taman Bungkul	Sedang	58.000
23	SD - 5	08/06/2017	Jl. Taman Mayangkara	Sedang	79.000
24	SD - 6	08/06/2017	Wonokromo	Sedang	100.000
25	SD - 1	09/06/2017	Jl. Raya Mulyosari	Murah	46.000
26	SD - 4	11/06/2017	Jl. Galaksi Klampis Asri Utara X	Murah	52.000
27	SD - 5	12/06/2017	Jl. Manyar Kertoarjo	Murah	53.000
14	SD - 6	14/06/2017	Jl. Raya Menur	Murah	60.000
15	SD - 2	10/06/2017	Jl. Galaksi Klampis Asri Utara X	Murah	52.000
16	SD - 3	12/06/2017	Jl. Manyar Kertoarjo	Murah	53.000

17	SD - 4	12/06/2017	Jl. Raya Menur	Murah	47.000
18	SD - 6	12/06/2017	Jl. Kedung Cowek	Murah	61.000
19	SD - 2	09/06/2017	Jl. Kedung Cowek	Sedang	61.000
20	SD - 4	11/06/2017	Taman Bungkul	Sedang	63.000
21	SD - 3	11/06/2017	Jl. Taman Mayangkara	Sedang	71.000
22	SD - 6	14/06/2017	Wonokromo	Sedang	71.000
23	SD - 4	02/06/2017	Jl. Raya Mulyosari	Sedang	46.000
24	SD - 5	03/06/2017	Jl. Galaksi Klampis Asri Utara X	Sedang	52.000
25	SD - 6	05/06/2017	Jl. Manyar Kertoarjo	Sedang	53.000
26	SMP - 7	04/06/2017	Jl. Raya Menur	Sedang	59.000
27	SD - 4	06/06/2017	Jl. Galaksi Klampis Asri Utara X	Sedang	52.000
14	SD - 5	07/06/2017	Jl. Manyar Kertoarjo	Sedang	53.000
15	SD - 6	07/06/2017	Jl. Raya Menur	Sedang	59.000
16	SMP - 8	06/06/2017	Jl. Kedung Cowek	Sedang	91.000
17	SD - 4	05/06/2017	Jl. Kedung Cowek	Mahal	61.000
18	SD - 5	04/06/2017	Taman Bungkul	Mahal	74.000
19	SD - 6	03/06/2017	Jl. Taman Mayangkara	Mahal	100.000

20	SMP - 9	02/06/2017	Wonokromo	Mahal	100.000
21	SD - 4	05/06/2017	Jl. Raya Mulyosari	Sedang	46.000
22	SD - 5	06/06/2017	Jl. Galaksi Klampis Asri Utara X	Sedang	52.000
23	SD - 6	07/06/2017	Jl. Manyar Kertoarjo	Sedang	53.000
24	SMP - 8	08/06/2017	Jl. Raya Menur	Sedang	59.000
25	SMP - 9	09/06/2017	Jl. Galaksi Klampis Asri Utara X	Sedang	54.000
26	SD - 4	09/06/2017	Jl. Manyar Kertoarjo	Sedang	53.000
27	SD - 5	08/06/2017	Jl. Raya Menur	Sedang	53.000
28	SD - 6	06/06/2017	Jl. Kedung Cowek	Sedang	91.000
29	SMP - 7	07/06/2017	Jl. Kedung Cowek	Mahal	93.000
30	SMP - 9	05/06/2017	Taman Bungkul	Mahal	94.000
31	SMP - 7	06/06/2017	Jl. Taman Mayangkara	Mahal	96.000
32	SMP - 8	07/06/2017	Wonokromo	Mahal	100.000
33	SD - 4	09/06/2017	Jl. Raya Mulyosari	Murah	46.000
34	SD - 5	10/06/2017	Jl. Galaksi Klampis Asri Utara X	Murah	53.000
35	SD - 6	12/06/2017	Jl. Manyar Kertoarjo	Murah	53.000
36	SMP - 7	13/06/2017	Jl. Raya Menur	Murah	60.000

37	SD - 4	13/06/2017	Jl. Galaksi Klampis Asri Utara X	Sedang	52.000
38	SD - 5	14/06/2017	Jl. Manyar Kertoarjo	Sedang	53.000
39	SD - 6	15/06/2017	Jl. Raya Menur	Sedang	60.000
40	SMP - 8	15/06/2017	Jl. Kedung Cowek	Sedang	61.000
41	SD - 4	16/06/2017	Jl. Kedung Cowek	Sedang	61.000
42	SD - 5	15/06/2017	Taman Bungkul	Sedang	58.000
43	SD - 6	14/06/2017	Jl. Taman Mayangkara	Sedang	71.000
44	SMP - 9	16/06/2017	Wonokromo	Sedang	71.000
45	SMP - 9	03/06/2017	Jl. Raya Mulyosari	Sedang	61.000
46	SMA - 10	04/06/2017	Jl. Galaksi Klampis Asri Utara X	Sedang	67.000
47	SMA - 11	05/06/2017	Jl. Manyar Kertoarjo	Sedang	69.000
48	SMA - 12	06/06/2017	Jl. Raya Menur	Sedang	82.000
49	UMUM - 13	07/06/2017	Jl. Galaksi Klampis Asri Utara X	Sedang	65.000
50	SMP - 9	07/06/2017	Jl. Manyar Kertoarjo	Sedang	53.000
51	SMA - 10	06/06/2017	Jl. Raya Menur	Sedang	79.000
52	SMA - 11	05/06/2017	Jl. Kedung Cowek	Sedang	92.000

53	SMA - 12	04/06/2017	Jl. Kedung Cowek	Mahal	93.000
54	UMUM - 13	03/06/2017	Taman Bungkul	Mahal	94.000
55	SMP - 9	02/06/2017	Jl. Taman Mayangkara	Mahal	100.000
56	SMA - 10	07/06/2017	Wonokromo	Mahal	94.000
57	SMP - 9	05/06/2017	Jl. Raya Mulyosari	Sedang	61.000
58	SMA - 10	06/06/2017	Jl. Galaksi Klampis Asri Utara X	Sedang	67.000
59	SMA - 11	07/06/2017	Jl. Manyar Kertoarjo	Sedang	69.000
60	SMA - 12	08/06/2017	Jl. Raya Menur	Sedang	88.000
61	UMUM - 13	09/06/2017	Jl. Galaksi Klampis Asri Utara X	Mahal	65.000
62	SMP - 9	09/06/2017	Jl. Manyar Kertoarjo	Mahal	53.000
63	SMA - 10	07/06/2017	Jl. Raya Menur	Mahal	79.000
64	SMA - 11	08/06/2017	Jl. Kedung Cowek	Mahal	92.000
65	SMA - 12	06/06/2017	Jl. Kedung Cowek	Mahal	91.000
66	UMUM - 13	05/06/2017	Taman Bungkul	Mahal	94.000
67	SMA - 11	09/06/2017	Jl. Taman Mayangkara	Mahal	97.000
68	UMUM - 13	08/06/2017	Wonokromo	Mahal	100.000
69	SMP - 9	09/06/2017	Jl. Raya Mulyosari	Sedang	61.000

70	SMA - 10	10/06/2017	Jl. Galaksi Klampis Asri Utara X	Sedang	50.000
71	SMA - 11	11/06/2017	Jl. Manyar Kertoarjo	Sedang	53.000
72	SMA - 12	12/06/2017	Jl. Raya Menur	Sedang	59.000
73	UMUM - 13	13/06/2017	Jl. Galaksi Klampis Asri Utara X	Sedang	54.000
74	SMP - 9	14/06/2017	Jl. Manyar Kertoarjo	Sedang	53.000
75	SMA - 10	15/06/2017	Jl. Raya Menur	Sedang	50.000
76	SMA - 11	16/06/2017	Jl. Kedung Cowek	Sedang	92.000
77	SMA - 12	16/06/2017	Jl. Kedung Cowek	Mahal	93.000
78	UMUM - 13	15/06/2017	Taman Bungkul	Mahal	94.000
79	SMA - 11	14/06/2017	Jl. Taman Mayangkara	Mahal	97.000
80	UMUM - 13	14/06/2017	Wonokromo	Mahal	100.000

5.4 Evaluasi

Pada **Tabel 5.4** dapat diamat bahwa terdapat 2 label merah yang berbeda. Pada label merah yang terdapat pada 1 baris kotak menandakan bahwa hasil dari estimasi harga tersebut dinyatakan salah karena hasil yang didapatkan lebih dari hasil maksimal yang telah ditentukan yaitu Rp 20.000. Untuk jenis label yang kedua dimana label merah hanya berada pada kategori harga dan hasil harga yang didapatkan menandakan bahwa harga yang didapatkan dinyatakan salah karena harga tersebut tidak termasuk kedalam kategori harga yang telah ditentukan sebelumnya.

Pada **Tabel 5.5** dapat diamati bahwa terdapat 2 label yang berbeda yaitu label kuning dan label merah. Label kuning menandakan bahwa hasil uji coba tersebut berkaitan dengan hasil uji coba yang ada pada **Tabel 5.4**. Sedangkan untuk label merah menyatakan bahwa terjadi kesalahan mengenai ketidak sesuaian selisih harga yang telah ditentukan dan ketidak sesuaian mengenai hasil harga yang didapatkan dengan kategori harga yang telah ditentukan pada aturan-aturan sebelumnya.

Maka dari itu, dapat diketahui bahwa hal utama yang mempengaruhi besarnya harga adalah tingkat kesulitan lalu diikuti oleh jarak dan kemudian waktu. Hal tersebut terjadi sesuai dengan aturan pemberian bobot yang telah dilakukan sebelumnya. Kesalahan pada hasil uji coba terjadi ketika varian inputan yang ada pada proses transaksi tidak sesuai dengan aturan yang telah dibangun didalam sistem dan adanya kesalahan batasan jarak yang terjadi karena *human error*.

Kendala yang ditemukan selama proses pengerjaan antara lain yaitu penentuan zona daerah untuk menangani ruang lingkup daerah Surabaya, memodelkan *base knowledge* yang didapatkan dari sistem pakar, pembobotan kriteria harga pada *rule* yang akan dibangun. Secara keseluruhan, hasil uji coba dapat diamat pada **Tabel 5.6** dan **Tabel 5.7**.

Tabel 5.6 Hasil Uji Coba Keseluruhan

Banyak Uji Coba	Benar	Salah	Akurasi
Selisih Estimasi Harga & Harga Akhir			
44	36	8	82%
Perhitungan Estimasi Harga			
44	36	8	82%
Perhitungan Harga Akhir			
80	66	14	82,5%

Tabel 5.7 Hasil Uji Coba Keseluruhan

No.	Kriteria	Valid
<i>Website</i>		
1.	Fitur Jumlah Tutor Baru	√
2.	Fitur Jumlah Murid Baru	√
3.	Fitur Jumlah Transaksi Baru	√
4.	Fitur Jumlah Pemasukan Baru	√
5.	Fitur Grafik Transaksi Per-Bulan	√
6.	Fitur Grafik Transaksi Per-Tahun	√
7.	Fitur Rekap Pendaftaran Tutor	√
8.	Fitur Rekap Pendaftaran Murid	√
9.	Fitur Rekap Tutor Terbaik	√
10.	Fitur Rekap Murid Terbaik	√
<i>Sistem Inferensi Fuzzy</i>		
1.	Pembatasan Tanggal H+14	√
2.	Pembatasan Jarak = 35.000 m	√
3.	Perhitungan Sesuai dengan Durasi	√
4.	Implementasi Zona Daerah	√
5.	Selisih estimasi harga dan harga akhir tidak lebih dari 20.000	√
6.	Harga Sesuai dengan <i>Rule</i> yang telah dibangun	√
7.	Harga Sesuai dengan Kategori Harga	√

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah. Selain itu juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut.

5.5 Kesimpulan

Kesimpulan yang diperoleh dari uji coba dan evaluasi adalah sebagai berikut:

1. Sistem Inferensi *Fuzzy Tsukamoto* berhasil diimplementasi pada proses perhitungan estimasi harga dan perhitungan harga akhir untuk aplikasi "*Finding-Tutor*" dengan rata-rata akurasi sebesar 82,2%.
2. Sistem Inferensi *Fuzzy (FIS) Tsukamoto* dapat menentukan jumlah harga sesuai dengan kriteria yang telah ditentukan yaitu tingkat kesulitan, waktu dan jarak.
3. Sistem Inferensi *Fuzzy (FIS) Tsukamoto* dapat menentukan jumlah harga dengan mempertimbangkan *fee* transportasi dan keadaan waktu diantara 2 pengguna.
4. Sistem Inferensi *Fuzzy (FIS) Tsukamoto* dapat menentukan jumlah harga sesuai dengan keadaan pasar saat ini.

5.6 Saran

Saran yang dapat diberikan dalam pengujian hasil implementasi adalah sebagai berikut:

1. Dapat menambah beberapa kriteria lain seperti keahlian untuk meningkatkan kualitas harga.
2. Dapat memodelkan representasi dari *base knowledge* lebih tepat lagi untuk mendapatkan hasil yang lebih maksimal.
3. Penentuan zona daerah yang lebih signifikan.
4. Peningkatan akurasi dengan cara evaluasi *rule* lebih dalam.

[Halaman ini sengaja dikosongkan]

LAMPIRAN

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] I. Wahyuni, W. F. Mahmudy, and A. Iriany, “Rainfall prediction in Tengger region Indonesia using Tsukamoto fuzzy inference system,” in *2016 1st International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, 2016, pp. 130–135.
- [2] J. O’Brien and G. Marakas, *Management Information Systems*, 10 edition. New York: McGraw-Hill Education, 2010.
- [3] T. J. Ross, *Fuzzy Logic with Engineering Applications, Third Edition*, 3 edition. Chichester, U.K: Wiley, 2010.
- [4] Fanoeel Thamrin, “STUDI INFERENSI FUZZY TSUKAMOTO UNTUK PENENTUAN FAKTOR PEMBEBANAN TRAFO PLN,” UNIVERSITAS DIPONEGORO, SEMARANG, 2012.
- [5] Diana Purwitasari, “Fuzzy Inference System (FIS),” Surabaya, Sep-2016.
- [6] “Java,” *Wikipedia bahasa Indonesia, ensiklopedia bebas*. 28-Jun-2016.
- [7] “NetBeans IDE - Overview.” [Online]. Available: <https://netbeans.org/features/>. [Accessed: 08-May-2017].
- [8] “Mengenal Android Studio | Android Studio.” [Online]. Available: <https://developer.android.com/studio/intro/index.html>. [Accessed: 08-May-2017].

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Syah Dia Putri Mustika Sari , lahir di Semarang, pada tanggal 25 Maret 1995. Penulis menempuh pendidikan mulai dari SDN Sompok 03 Semarang (2001-2007), SMPN 21 Semarang (2007-2010), SMAN 4 Semarang (2010-2013) hingga terakhir Institut Teknologi Sepuluh Nopember Surabaya (2013-2017) di jurusan Teknik Informatika, Fakultas Teknologi Informasi angkatan tahun 2013.

Selama belajar di kampus Teknik Informatika, penulis berkesempatan menjadi asisten dosen Sistem Teknologi Informasi pada tahun 2016. Selain mengikuti kegiatan akademik, penulis mengikuti kegiatan organisasi sekaligus mengembangkan minatnya dengan berpartisipasi sebagai anggota Departemen Hubungan Luar HMTTC (2014-2015), Badan Pengurus Harian (Wakil Koordinator) REEVA Schematics (2015) dan Pemandu FTIf (2014-2016).

Penulis memiliki bidang minat Manajemen Informasi (MI) dengan fokus studi pada bidang *mobile application* dan *website*. Komunikasi dengan penulis mengenai tugas akhir dapat melalui email: **syahdiaputri@gmail.com**.